

KONZEPTION UND ENTWICKLUNG EINES  
DSM-BASIERTEN GAMIFICATION AUTHORIZING  
SYSTEMS ZUR UNTERSTÜTZUNG  
HOCHSCHULISCHER LEHRE



Inaugural-Dissertation zur Erlangung der Doktorwürde  
der Fakultät für Sprach-, Literatur- und Kulturwissenschaften  
der Universität Regensburg

Vorgelegt von  
Alexander Bartel  
aus  
Kempten

2018

Alexander Bartel, M.Eng.: *Konzeption und Entwicklung eines DSM-basierten Gamification Authoring Systems zur Unterstützung hochschulischer Lehre*

ERSTGUTACHTER: Prof. Dr. phil. Christian Wolff, Universität Regensburg

ZWEITGUTACHTER: Prof. Dr. rer. nat. Georg Hagel, Hochschule Kempten

TAG DER ABGABE: 30. Mai 2018

TAG DER DISPUTATION: 29. November 2018

KOOPERATION: Die Arbeit entstand in gemeinsamer Betreuung durch die Fakultät für Sprach-, Literatur- und Kulturwissenschaften der Universität Regensburg und die Informatik-Fakultät der Hochschule Kempten.

Regensburg, 2018



---

## DANKSAGUNG

---

Einen herzlichen Dank möchte ich an meine Betreuer und Gutachter Prof. Dr. Georg Hagel sowie Prof. Dr. Christian Wolff richten. Während der Anfertigung dieser Arbeit wurde ich immer mit wertvollem Rat unterstützt und stets mit einem freundlichen und offenen Ohr empfangen. Durch die Diskussionen mit ihnen habe ich sicherlich nicht nur Dinge für diese Arbeit gelernt. Ebenso bedanke ich mich bei Prof. Dr. Kersten Reich für die Begutachtung dieser Arbeit.

Mein Dank geht ebenso an meine Kolleginnen und Kollegen der Hochschule Kempten und des Projekts EVELIN. Ebenso danke ich allen Teilnehmerinnen und Teilnehmern, die an den in dieser Arbeit durchgeführten Studien teilnahmen, für ihre Zeit und ihren wertvollen Input. Darüber hinaus bedanke ich mich bei den Korrekturlesern dieser Arbeit, allen voran bei Stefan Frenz.

Ich möchte meinen Eltern danken, für ihre selbstlose Unterstützung über die Jahre und die Ermöglichung von so Vielem.

Weiterhin danke ich all meinen Freunden für ihre Geduld aufgrund meiner Höhen und Tiefen während dieser Zeit sowie für ihr Verständnis für das Fernbleiben an diversen Abenden.

Ein besonderer Dank gilt meiner Frau Paula, der ich diese Arbeit widme. Ich danke ihr für ihre fortwährende und unendlich liebevolle Unterstützung. Ich bin dankbar, dass es dich gibt.

---

## ABSTRACT

---

The overall aim of the present work is to create a systematization that enables an efficient conception and implementation of gamified teaching or rather learning units for and in higher education and to implement this systematization as a prototype. Therefore, the current state of research on gamification in higher education teaching is captured from a content-structural and motivation-psychological perspective. Based on this, a domain analysis is carried out, which consists of a document analysis and learning platform analysis, with the aim of extracting established domain concepts. The resulting findings serve as the basis for the conception and development of a domain-specific modeling language as part of the domain-specific modeling approach. The implementation of other components of the approach are also detailed, such as a designer that allows to create models based on the language, a generator that transforms the created models into code, as well as a learning management system in which the models available in code are used. Furthermore, an evaluation is described, consisting of a qualitative field study and a comparison of requirements. Additionally potential areas of application for the developed artifacts are exemplarily shown.

---

## KURZZUSAMMENFASSUNG

---

Vorliegende Arbeit hat das Ziel eine Systematisierung zu erstellen, die eine effiziente Konzeption und Implementierung von spielifizierten Lehr-Lerneinheiten für und in die Hochschullehre ermöglicht und diese Systematisierung prototypisch umzusetzen. Hierfür wird zunächst der aktuelle Stand der Forschung zu Gamification in der Hochschullehre aus inhaltlich-struktureller und motivationspsychologischer Perspektive erfasst. Darauf aufbauend wird eine Domänenanalyse durchgeführt, welche aus einer Dokumentenanalyse und Lernplattformanalyse besteht, mit dem Ziel, etablierte Domänenkonzepte zu exzerpieren. Die daraus gewonnenen Erkenntnisse dienen als Basis für die Konzeption und Entwicklung einer domänenspezifischen Modellierungssprache als Teil des Domain-specific Modeling Ansatzes. Die Umsetzung weiterer Bestandteile des Ansatzes, wie einen Designer, der es erlaubt Modelle auf Basis der Sprache zu erstellen, einen Generator, der erstellte Modelle in Code transformiert, sowie ein Learning Management System, in dem die in Code vorliegenden Modelle zum Einsatz kommen, werden ebenso detailliert. Weiterhin wird eine Evaluation beschrieben, bestehend aus einer qualitativen Feldstudie und einem Abgleich von Anforderungen. Darüber hinaus werden potentielle Einsatzgebiete für die entwickelten Artefakte beispielhaft aufgezeigt.

---

## INHALTSVERZEICHNIS

---

1	EINFÜHRUNG	1
1.1	Problemkontext und wissenschaftlicher Beitrag . . . . .	1
1.2	Forschungsdesign . . . . .	4
1.3	Organisatorischer Rahmen . . . . .	8
1.4	Struktur der Arbeit . . . . .	8
2	GAMIFICATION - DIE ANATOMIE DES ANSATZES	11
2.1	Hinführung . . . . .	11
2.2	Definitorischer Rahmen . . . . .	13
2.2.1	Begriffsheterogenität . . . . .	15
2.2.2	Definition für diese Forschungsarbeit . . . . .	16
2.3	Verortung des Gamification-Begriffs . . . . .	17
2.3.1	Serious Game Design . . . . .	18
2.3.2	Serious Toy Design . . . . .	20
2.3.3	Gameful Design . . . . .	21
2.3.4	Playful Design . . . . .	23
2.3.5	Synonymische Verwendungen . . . . .	24
2.4	Gamification Design-Elemente . . . . .	25
2.4.1	Begriffliche Annäherung . . . . .	26
2.4.2	Klassifikationsschemata . . . . .	27
2.5	Kompetenzentwicklung durch Gamification Design-Elemente . .	36
2.5.1	Merkmale kompetenzorientierter Lehre . . . . .	36
2.5.2	Dynamics als Brücke zur Kompetenzorientierung . . . . .	38
2.5.3	Validierung der Theorie . . . . .	43
2.6	Zwischenfazit . . . . .	58
3	MOTIVATIONSFÖRDERUNG DURCH GAMIFICATION	59
3.1	Einleitung . . . . .	59
3.2	Motivationstheoretische Grundlagen . . . . .	61
3.2.1	Begriffsannäherung . . . . .	61
3.2.2	Zum Motivationsbegriff . . . . .	61
3.2.3	Intrinsische und extrinsische Motivation . . . . .	63
3.2.4	Einschlägige Motivationstheorien . . . . .	65

3.3	Zur Lernmotivation . . . . .	79
3.3.1	Lernmotivation und deren Bedeutung . . . . .	80
3.3.2	Ein Rahmenmodell zur Lernmotivation nach Rheinberg und Fries	81
3.3.3	Förderung von Lernmotivation . . . . .	85
3.3.4	Designprinzipien zur Förderung von Lernmotivation durch Ga- mification . . . . .	90
3.4	Zwischenfazit und Forschungsdesiderate . . . . .	99
4	EMPIRISCHE STUDIEN ZU GAMIFICATION IN DER LEHRE	101
4.1	Einleitung . . . . .	101
4.2	Dokumentenanalyse zu Gamification Design-Elementen in der Lehre . . . . .	102
4.2.1	Verwandte Untersuchungen . . . . .	105
4.2.2	Methodentheoretische Vorgangsbeschreibung . . . . .	116
4.2.3	Instanziierung der beschriebenen Methodik . . . . .	127
4.2.4	Zwischenergebnis - Dokumentenanalyse . . . . .	138
4.2.5	Maßnahmen zur Qualitätssicherung . . . . .	147
4.3	Lernplattformanalyse . . . . .	149
4.3.1	Identifikation von spielifizierten Lernplattformen . . . . .	150
4.3.2	Inklusionskriterien . . . . .	150
4.3.3	Schema zur Untersuchung von Gamification Design-Elementen	152
4.3.4	Durchführung . . . . .	155
4.3.5	Zwischenergebnis - Lernplattformanalyse . . . . .	157
4.4	Zwischenfazit . . . . .	166
5	EMENDO DOMAIN-SPECIFIC MODELING LANGUAGE	171
5.1	Hinführung . . . . .	171
5.2	Grundlagen einer Domain-specific Modeling Language . . . . .	172
5.2.1	Allgemeine Merkmale von Sprachen . . . . .	172
5.2.2	Modellierungssprachen . . . . .	173
5.2.3	Domänenspezifische Modellierungssprachen . . . . .	174
5.3	Bestandteile von Domain-specific Modeling . . . . .	178
5.3.1	Sprache . . . . .	179
5.3.2	Modelle . . . . .	186
5.3.3	Generatoren . . . . .	187
5.3.4	Domain Framework . . . . .	191
5.4	Vorüberlegung zur Erstellung der DSML . . . . .	192
5.4.1	Vorgehen . . . . .	192
5.4.2	Zweck und Anwendungsbereich . . . . .	194

5.4.3	Anforderungen an die Emendo DSML . . . . .	195
5.5	Abstrakte Syntax . . . . .	197
5.5.1	EmendoModell . . . . .	199
5.5.2	Verbindung und Gruppe . . . . .	199
5.5.3	Lernelement und Lernbaustein . . . . .	200
5.5.4	Lernmaterial . . . . .	201
5.5.5	Aufgabe . . . . .	202
5.5.6	Regel . . . . .	204
5.6	Konkrete Syntax . . . . .	216
5.6.1	Nutzerstudie . . . . .	216
5.6.2	Umsetzung der konkreten Syntax . . . . .	217
5.7	Zwischenfazit . . . . .	220
6	EMENDO DESIGNER UND GENERATOR . . . . .	221
6.1	Einleitung . . . . .	221
6.2	Eclipse Modeling Framework . . . . .	222
6.2.1	Ecore und seine Sprachkonzepte . . . . .	222
6.2.2	Umsetzung der abstrakten Syntax in Ecore . . . . .	224
6.2.3	Generierung von Eclipse Plug-In-Projekten . . . . .	225
6.3	Sirius . . . . .	233
6.3.1	Systembestandteile . . . . .	233
6.3.2	Erstellung des Emendo Designers mit Sirius . . . . .	234
6.3.3	GUI-Überblick . . . . .	235
6.4	Relevante Modellierungsaspekte des Emendo Designers . . . . .	237
6.4.1	Korrektheitsüberprüfungen bei Zuordnungsverbindungen . . . . .	237
6.4.2	Korrektheitsüberprüfungen bei Pflicht- und Optionalverbindungen . . . . .	239
6.4.3	Syntaktische Validierung von Bezeichnern . . . . .	240
6.4.4	Automatisches Setzen von Attributwerten . . . . .	240
6.4.5	Kontextabhängige Definition von Objektzuständen über Eingabedialoge . . . . .	242
6.4.6	Modellweite Synchronisation von Punkte- und Levelsystem . . . . .	244
6.5	Emendo Generator . . . . .	246
6.5.1	Integration in den Emendo Designer . . . . .	246
6.5.2	Struktur des Generators . . . . .	247
6.5.3	Anbindung an Domain Framework . . . . .	248
6.6	Verwandte Arbeiten . . . . .	249
6.6.1	Game Description Language und ihre Weiterentwicklung . . . . .	250
6.6.2	Card Game Description Language . . . . .	251

6.6.3	Gamification Modeling Language . . . . .	252
6.6.4	Machinations . . . . .	254
6.6.5	ATTAC-L . . . . .	256
6.6.6	Weitere relevante Arbeiten . . . . .	258
6.7	Zwischenfazit . . . . .	261
7	EMENDO LEARNING MANAGEMENT SYSTEM . . . . .	262
7.1	Einleitung . . . . .	262
7.2	Anforderungsanalyse . . . . .	262
7.2.1	Funktionale und nichtfunktionale Anforderungen . . . . .	263
7.2.2	Use Cases des Emendo LMS . . . . .	264
7.3	Systemarchitektur . . . . .	267
7.3.1	Model-View-Controller als Architekturmuster für das Emendo LMS . . . . .	267
7.3.2	PanelHandler . . . . .	269
7.3.3	LernelementHandler . . . . .	270
7.3.4	RegelHandler . . . . .	275
7.3.5	Datenbank . . . . .	286
7.4	Grafische Benutzerschnittstelle . . . . .	288
7.4.1	Hauptpanels . . . . .	289
7.4.2	Bearbeitung von Lernbausteinen . . . . .	291
7.4.3	Überwachung des Lernfortschritts und Lehrendenfeedbacks . . . . .	294
7.5	Mobile Emendo LMS . . . . .	295
7.6	Zwischenfazit . . . . .	297
8	EVALUATION . . . . .	298
8.1	Einleitung . . . . .	298
8.2	Evaluation der DSML und des Designers . . . . .	299
8.2.1	Ziele und Vorüberlegungen . . . . .	299
8.2.2	Methodische Vorüberlegungen . . . . .	300
8.2.3	Durchführung . . . . .	301
8.2.4	Ergebnisse . . . . .	308
8.2.5	Diskussion . . . . .	319
8.2.6	Methodische Kritik und Grenzen . . . . .	326
8.3	Evaluation der Anforderungen an die DSML . . . . .	327
8.3.1	Semantische und grafische Affordance . . . . .	327
8.3.2	Semantische Eindeutigkeit . . . . .	328
8.3.3	Erweiterbarkeit . . . . .	328
8.3.4	Separierte Abstraktionslevel . . . . .	329

8.3.5	Eindeutigkeit von Sprachkonzept zu Zielkonzept . . . . .	329
8.4	Emendo für spielifiziertes Just-in-Time Teaching . . . . .	330
8.4.1	JiTT als Lehr-Lernkonzept . . . . .	330
8.4.2	Ablaufbeschreibung . . . . .	331
8.4.3	Spielifiziertes JiTT . . . . .	336
8.4.4	Beispielhafte Umsetzung einer spielifizierten JiTT-Lerneinheit .	337
8.5	Zwischenfazit . . . . .	344
9	ZUSAMMENFASSUNG UND AUSBLICK	345
9.1	Zusammenfassung der wesentlichen Ergebnisse dieser Arbeit . .	345
9.1.1	Betrachtung von Gamification und Potentiale für die kompeten-	
	tenzororientierte Lehre . . . . .	345
9.1.2	Motivationstheoretische Sicht auf Gamification . . . . .	346
9.1.3	Empirische Studien zu Gamification in der Lehre . . . . .	347
9.1.4	Emendo DSML . . . . .	348
9.1.5	Emendo Designer und Generator . . . . .	349
9.1.6	Emendo Learning Management System . . . . .	350
9.1.7	Evaluation . . . . .	351
9.2	Ausblick und weitere Forschungsarbeiten . . . . .	351
9.2.1	Portierung als Webanwendung . . . . .	352
9.2.2	Weiterentwicklung der Notation des Emendo Designer bzw. der	
	Emendo DSML . . . . .	352
9.2.3	Anbindung an bestehende Learning Management Systeme . . .	353
9.2.4	Adaptivität des Emendo LMS . . . . .	354
	LITERATURVERZEICHNIS	355
A	ANHÄNGE	403
A.1	Der Gamified Course Design Record . . . . .	404
A.2	Schema zur Lernplattformanalyse . . . . .	407
A.3	Fragebogen zur Findung von Notationsvorschlägen für die Emen-	
	do DSML . . . . .	408
A.4	Verbindungsübersicht über Notationselemente . . . . .	415
A.5	Anforderungen an das Emendo LMS . . . . .	416
A.6	Screenshots Hauptseiten des Emendo LMS . . . . .	423
A.7	Screenshots des Emendo LMS Mobile . . . . .	430
A.8	Geheft für die zweite Evaluationsstufe . . . . .	435
A.9	Fragenbogen für Evaluationsteilnehmende . . . . .	439



---

## ABBILDUNGSVERZEICHNIS

---

Abbildung 1	Komplementarität von Design Science und Behavioral Science nach Hevner u. Chatterjee (2010, S. 11) . . . . .	5
Abbildung 2	Information System Research Framework, eigene Übersetzung nach Hevner et al. (2004, S. 80) . . . . .	6
Abbildung 3	Einordnung von Gamification in die angrenzende Begriffswelt (vgl. Deterding (2016, S. 105)) . . . . .	18
Abbildung 4	Screenshot aus SimSE (Navarro, 2006, S. 104) . . . . .	20
Abbildung 5	LEGO Mindstorms EV3 Affe . . . . .	22
Abbildung 6	Screenshot der Lernplattform QuestTanja (Stöcklin et al., 2014, S. 272) . . . . .	23
Abbildung 7	Playful Design bei Alltagsgegenständen (Grotton, 2012)	24
Abbildung 8	Google-Suchanfragen zu Gamification und synonymisch verwendeten Begriffen im zeitlichen Verlauf (Jan. 2004 - Jul. 2015) . . . . .	25
Abbildung 9	Schells Tetrade der Game-Elemente (Schell, 2015, S. 51)	30
Abbildung 10	Hunicke's Mechanics-Dynamics-Aesthetics-Framework (vgl. Hunicke et al., 2004, S. 2) . . . . .	32
Abbildung 11	Gamification Toolkit nach Werbach u. Hunter (2015, S. 22ff.) . . . . .	34
Abbildung 12	Zusammenhang von Dynamics, Mechanics und Components (Auszug aus Bartel u. Hagel (2016b, S. 75) bzw. Werbach u. Hunter (2015, S. 22)) . . . . .	35
Abbildung 13	Der GCDP von Bartel u. Hagel (2016b, S. 75) . . . . .	44
Abbildung 14	Determinanten motivationalen Handelns nach (Figas et al., 2013, S. 1300, eigene Übersetzung) bzw. Gage u. Berliner (1996, S. 338) . . . . .	63
Abbildung 15	Flow Theory nach Csikszentmihalyi (2010, S. 74, eigene Darstellung) . . . . .	67
Abbildung 16	SDT nach Deci u. Ryan (1985) . . . . .	70
Abbildung 17	Theory of Planned Behavior (eigene Darstellung nach Ajzen (1991, S. 182)) . . . . .	76
Abbildung 18	Modell der Schlüsselprozesse bei Verhaltensinterventionen (eigene Darstellung nach Connor et al. (2014, S. 97)) . . . . .	77

Abbildung 19	Rahmenmodell zur Lernmotivation und deren Auswirkungen nach (Rheinberg, 1986, S. 170) . . . . .	81
Abbildung 20	Modell zur Förderung der Lernmotivation im Software Engineering nach (Figas et al., 2013, S. 1302, eigene Übersetzung) . . . . .	87
Abbildung 21	Schematischer Ablauf der Domänenanalyse . . . . .	102
Abbildung 22	Word Cloud über Konzepte und Ideen zu Gamification in der Lehre (Caponetto et al., 2014, S. 53) . . . . .	109
Abbildung 23	Klassifizierungsschema von Gamification-Literatur im Bereich der Informationssysteme nach Schlagenhauer u. Amberg (2015, S. 6) . . . . .	112
Abbildung 24	Ablaufmodell strukturierender qualitativer Inhaltsanalyse (Mayring, 2015, S. 98) . . . . .	127
Abbildung 25	Beispielhafte Abstraktion von Textsegmenten zu konzeptuellen Anforderungen von Konzepten . . . . .	140
Abbildung 26	Beispiele für Badges in Form von Medaillen in einem LMS (Dominguez et al., 2013, S. 384) . . . . .	141
Abbildung 27	Beispielhafte Visualisierung der GDE, CC und Lernelemente in einem Übungskurs der Lernplattform <i>Classgame</i> (Fuchs, 2016) . . . . .	157
Abbildung 28	Beispielhafte Darstellung der vier identifizierten Aufgabentypen . . . . .	162
Abbildung 29	Problem- und Lösungsdomäne im Kontext von DSM . . . . .	174
Abbildung 30	Übergang von Domänenidee bis zum fertigen Produkt - Traditionelle Softwareentwicklung und DSM im Vergleich (Kelly u. Tolvanen, 2008, S. 16) . . . . .	176
Abbildung 31	Relativer Vergleich domänenspezifische Modellierungssprachen mit weiteren Sprachtypen (auf Basis von Kleppe (2009, S. 32)) . . . . .	178
Abbildung 32	Beispiele für eine textuelle und grafische konkrete Syntax . . . . .	180
Abbildung 33	Beziehung zwischen Klasse und Attribut in einem Meta-Metamodell (Auszug aus OMG (2014, S. 42)) . . . . .	182
Abbildung 34	Überblick über die Bestandteile von Emendo Domain-Specific Modeling (DSM) . . . . .	195
Abbildung 35	Das Metamodell der Emendo DSML . . . . .	198
Abbildung 36	Verbindungen in Emendo Domain-Specific Modeling Language (DSML) . . . . .	200

Abbildung 37	Metamodell inkl. konkreter Ausprägungen zu Lernmaterialien . . . . .	202
Abbildung 38	Metamodell zu Ausprägungen von Aufgaben . . . . .	203
Abbildung 39	Auf Lernbausteine bezogene Benutzeraktionen in Emendo DSML . . . . .	206
Abbildung 40	Auf Gruppen bezogene Benutzeraktionen in Emendo DSML . . . . .	208
Abbildung 41	Ausdrücke von Regeln der Emendo DSML . . . . .	209
Abbildung 42	Boolsche Bedingungsvariablen . . . . .	210
Abbildung 43	Klassendiagramm der Aufgabentyp Bedingungsvariable zum Vergleich von Aufgabentyp-Ausdrücken . . . . .	211
Abbildung 44	Ganzzahl Bedingungsvariablen . . . . .	212
Abbildung 45	Feedback- und Belohnungskonsequenzen . . . . .	213
Abbildung 46	Game Design-Elemente - Components . . . . .	214
Abbildung 47	Entwicklung der Notationselemente der Emendo DSML am Beispiel einer Single Choice-Aufgabe . . . . .	217
Abbildung 48	Konkrete Syntax der Emendo DSML . . . . .	218
Abbildung 49	Auszug aus Emendo DSML Metamodell - Beziehung von Hinweis zu Aufgabe . . . . .	222
Abbildung 50	Ein- und Ausgabe von Ecore (Steinberg et al., 2009, S. 21) . . . . .	222
Abbildung 51	Überblick über wesentliche Sprachkonzepte von Ecore nach (Brambilla et al., 2017, S. 101) . . . . .	223
Abbildung 52	Hierarchische Darstellung der Systemarchitektur eines auf Eclipse basierenden Modellierungswerkzeuges (Viyović et al., 2014, S. 234) . . . . .	233
Abbildung 53	Struktur des Emendo DSML Viewpoint Specification Models . . . . .	235
Abbildung 54	GUI des Emendo Designers . . . . .	236
Abbildung 55	Visuelle Rückmeldung der Überprüfungen bei der Erstellung von Zuordnungsverbindungen . . . . .	238
Abbildung 56	Vermeidung von symmetrischen zyklischen Abhängigkeiten bei Pflichtverbindungen . . . . .	239
Abbildung 57	Zuordnung von Hinweis zu Eingabe-Aufgabe . . . . .	240
Abbildung 58	Beispiele für JFace-Dialoge zur Detaillierung von Objektinformationen . . . . .	243
Abbildung 59	Auszug aus der Syntax der CGDL bei der Definition von UNO (Font et al., 2013, S. 261) . . . . .	252

Abbildung 60	Machinations Diagramm eines Strategiespiels (Dormans, 2012, S. 203) . . . . .	255
Abbildung 61	Beispiel für ein ATTAC-L Storyline Modell (de Troyer et al., 2017, S. 577) . . . . .	257
Abbildung 62	Auszug: Annotation eines Game Moves mit Scores (de Troyer et al., 2017, S. 578) . . . . .	258
Abbildung 63	Sketch eines prototypischen spielifizierten Modellierungseeditors (Yohannis, 2016, S. 5) . . . . .	259
Abbildung 64	Use Case Diagramm Emendo Learning Management System (LMS) . . . . .	265
Abbildung 65	Die Komponente PanelHandler . . . . .	269
Abbildung 66	Die Komponente LernelementHandler . . . . .	271
Abbildung 67	Sequenzdiagramm - Erstellung einer Single Choice-Aufgabe gemäß dem Construction Builder-Muster . . . . .	272
Abbildung 68	Zustandsdiagramm Aufgabe und Aufgabenhinweis . . . . .	274
Abbildung 69	Zustandsdiagramm für die Lernmaterialien Podcast, Video und Textblock . . . . .	274
Abbildung 70	Die Komponente Lernelement . . . . .	276
Abbildung 71	Die Komponente RegelHandler . . . . .	277
Abbildung 72	Interne Datenrepräsentation von vier Regel-Ausdrücken in einer String-Liste einer Emendo Modell Instanz inkl. Legende . . . . .	278
Abbildung 73	Aktivitätsdiagramm - Ausdruck parsen . . . . .	279
Abbildung 74	Aktivitätsdiagramm - Bedingung parsen . . . . .	280
Abbildung 75	Aktivitätsdiagramm - Regeln für Lernbaustein überprüfen . . . . .	282
Abbildung 76	Aktivitätsdiagramm - Regel evaluieren . . . . .	283
Abbildung 77	Aktivitätsdiagramm - Bedingung interpretieren . . . . .	284
Abbildung 78	SQL-Datenbankmodell von Emendo LMS . . . . .	287
Abbildung 79	Papierprototyp der Emendo LMS Bildschirmseite Home . . . . .	289
Abbildung 80	Screenshot der Lernbausteinübersicht im Emendo LMS . . . . .	290
Abbildung 81	Screenshot der Profilseite im Emendo LMS . . . . .	290
Abbildung 82	Benachrichtigung für den Erhalt eines Badges . . . . .	292
Abbildung 83	Screenshots des Emendo LMS Mobile – Single Choice-Aufgabe und Freitextaufgabe . . . . .	296
Abbildung 84	Beispielhafte Fotodokumentation eines Ergebnisses zum ersten Evaluationsteil . . . . .	306
Abbildung 85	Übersicht über die Codes der Stimuli-Frage von FFE_1 . . . . .	309

Abbildung 86	Schlüsselkategorien und Codes zum Konzept Einsetzbarkeit der FFE_1 . . . . .	310
Abbildung 87	Schlüsselkategorien und Codes zum Konzept Voraussetzungen der FFE_2 . . . . .	313
Abbildung 88	Übersicht über die Codes der Stimuli-Frage von FFE_3	315
Abbildung 89	Schlüsselkategorien und Codes zum Konzept Sprache der FFE_3 . . . . .	315
Abbildung 90	Schlüsselkategorien und Codes zum Konzept Sprache der FFE_4 . . . . .	316
Abbildung 91	Schlüsselkategorien und Codes zum Konzept Designer der FFE_5 . . . . .	318
Abbildung 92	Beispielfolie einer JiTT-Lehrveranstaltung . . . . .	332
Abbildung 93	Beispiel eines JiTT Puzzles (Patterson, 1999) . . . . .	335
Abbildung 94	Modellbeispiel zu einer spielifizierten JiTT-Einheit zum Thema Zustandsmuster . . . . .	339
Abbildung 95	Laserbar aus initialen Erzählung der spielifizierten JiTT-Einheit . . . . .	340
Abbildung 96	Screenshot Bildschirmseite Home . . . . .	423
Abbildung 97	Screenshot Bildschirmseite Lernbausteine . . . . .	423
Abbildung 98	Screenshot Bildschirmseite Profil . . . . .	424
Abbildung 99	Screenshot Bildschirmseite Bestenliste . . . . .	424
Abbildung 100	Screenshot Bildschirmseite Diskussion . . . . .	425
Abbildung 101	Screenshot Bearbeitung Eingabe-Aufgabe . . . . .	425
Abbildung 102	Screenshot Bearbeitung Freitext-Aufgabe . . . . .	426
Abbildung 103	Screenshot Bearbeitung Multiple Choice-Aufgabe . . . . .	426
Abbildung 104	Screenshot Bearbeitung Single Choice-Aufgabe . . . . .	427
Abbildung 105	Screenshot Bearbeitung eines Textblocks . . . . .	427
Abbildung 106	Screenshot Bearbeitung eines Podcasts . . . . .	428
Abbildung 107	Screenshot Bearbeitung eines Videos . . . . .	428
Abbildung 108	Screenshot Dozierendenansicht zum Lernfortschritt . . . . .	429
Abbildung 109	Screenshot Dozierendenansicht manuelle Bewertung . . . . .	429
Abbildung 110	Screenshots des Emendo LMS Mobile – Hauptseiten . . . . .	430
Abbildung 111	Screenshots des Emendo LMS Mobile – Hauptseiten Fortsetzung . . . . .	431
Abbildung 112	Screenshots des Emendo LMS Mobile – Aufgabentypen . . . . .	432
Abbildung 113	Screenshots des Emendo LMS Mobile – Lernmaterialien . . . . .	433

Abbildung 114	Screenshots des Emendo LMS Mobile – Dozentenansicht . . . . .	434
---------------	---	-----

---

## TABELLENVERZEICHNIS

---

Tabelle 1	Taxonomie von Game Design-Elementen . . . . .	29
Tabelle 2	Inhaltliche Beschreibungselemente von Dynamics . .	45
Tabelle 3	Übersicht über für Gamification relevante psychologische Theorien (angelehnt an Schlagenhauser u. Amberg (2014, S. 7) und Woolfolk (2014, S. 390)) . . . . .	66
Tabelle 4	Beispiele für die Gegensätzlichkeit von Kontroll- und Informationsaspekte in der kongitiven Bewertungstheorie (Woolfolk, 2014, S. 393) . . . . .	72
Tabelle 5	SMART-Zieleigenschaften im Kontext spielifizierten Lernens . . . . .	92
Tabelle 6	Auswertungskriterien und Bedeutung nach de Sousa Borges et al. (2014, S. 218, eigene Übersetzung) . . . .	107
Tabelle 7	Übersicht über vorgestellte Literature Reviews zum Thema Gamification in der Lehre (x = Nennung und Diskussion, (x) = Nennung, o = keine Nennung) . . . .	115
Tabelle 8	Taxonomie von Literature Reviews nach Cooper (1988)	117
Tabelle 9	Instanz der Taxonomie von Literature Reviews nach Cooper (1988) . . . . .	131
Tabelle 10	Ausgewählte elektronische Datenbanken für die Literatursuche . . . . .	133
Tabelle 11	Übersicht über Querystrings . . . . .	134
Tabelle 12	Ergebnisübersicht der Literatursuche . . . . .	137
Tabelle 13	Übersicht über codierte Arbeiten pro Typ . . . . .	138
Tabelle 14	Untersuchte spielifizierte Lehr-Lernplattformen . . . .	151
Tabelle 15	Übersicht über identifizierte Elemente aus Dokumentenanalyse und Lernplattformanalyse . . . . .	167
Tabelle 16	Abstraktionsstufen einer MOF Metamodellarchitektur (angelehnt an OMG (2014) bzw. Atkinson (1999, S. 24f.)) . . . . .	183
Tabelle 17	Beispielhafte Metamodell-Formalismen für SQL und XML . . . . .	184
Tabelle 18	Anwendungsfallbeschreibung Podcast bearbeiten . . .	267
Tabelle 19	Übersicht über das Sample inkl. Merkmalsausprägungen	304

Tabelle 20	Leitfragen für die inhaltliche Erhebungsphase der ersten Evaluationsstufe . . . . .	305
Tabelle 21	Leitfragen für die inhaltliche Erhebungsphase der zweiten Evaluationsstufe . . . . .	307



---

## CODEBEISPIELVERZEICHNIS

---

Code 1	Beispiel für die Generierung von Java-Quellcode mit Hilfe eines in Java geschriebenen Programms . . . . .	189
Code 2	Auszug aus einer Acceleo-Defintion zur Generierung von Java-Code . . . . .	190
Code 3	Java Code Output des Beispieltemplates . . . . .	191
Code 4	Darstellung der Klasse Hinweis in Ecore . . . . .	225
Code 5	Hinweis als Referenz in Aufgabe in Ecore . . . . .	225
Code 6	Generiertes Java Interface zur Ecore-Repräsentation der Klasse Hinweis . . . . .	226
Code 7	Generierte Java Interface-Implementierungsklasse zur Ecore-Repräsentation der Klasse Hinweis . . . . .	227
Code 8	Generierter Java Itemprovider der Klasse Hinweis . . .	230
Code 9	AQL-Constraints zur Erstellung einer Zuordnungsverbindung zwischen zwei Notationselementen . . . . .	238
Code 10	Konkreter ModelChangeTrigger zur Synchronisation von Punktesystem-Objekten . . . . .	245
Code 11	Template Auszug aus dem Hauptmodul von des Emen- do Generators zur Generierung einer Belohnungsregel	248
Code 12	Formale Beschreibung von Schach mit GDL-II (vollständige Information) . . . . .	250
Code 13	Formale Beschreibung von Schach mit GDL-II (unvollständige Information) . . . . .	250
Code 14	Beispielhaftes textuelles GaML-Modell eines Carpools	253
Code 15	OCL-Ausdruck zur Überprüfung von Modellierungsaktionen von Lernenden . . . . .	260
Code 16	Implementierte Methode zum Interpretieren eines Ausdrucks aus der LMS Bedingungsklasse Ungleich . . .	285
Code 17	Beispielhafte Methode des Persistence Repositories von EingabeAufgaben . . . . .	288
Code 18	An C angelehnter Pseudocode-Algorithmus zum Ermitteln des nächsten bearbeitbaren Lernbausteins . . .	292

---

## AKRONYME

---

<b>API</b>	Application Programming Interface
<b>AQL</b>	Acceleo Query Language
<b>BMBF</b>	Bundesministerium für Bildung und Forschung
<b>BNF</b>	Backus-Naur-Form
<b>CC</b>	Component Class
<b>CEGE</b>	Core Elements of Game Experience
<b>CGDL</b>	Card Game Description Language
<b>CNL</b>	Controlled Natural Language
<b>CRUD</b>	Create-Read-Update-Delete
<b>CSSL</b>	Computer-Supported Collaborative Learning
<b>CSS</b>	Cascading Style Sheets
<b>DAO</b>	Data Access Object
<b>DBS</b>	Datenbanksystem
<b>DDL</b>	Data Definition Language
<b>DCL</b>	Data Control Language
<b>DML</b>	Data Manipulation Language
<b>DSL</b>	Domain-Specific Language
<b>DSM</b>	Domain-Specific Modeling
<b>DSML</b>	Domain-Specific Modeling Language
<b>DSR</b>	Design Science Research
<b>DOM</b>	Document Object Model
<b>EMF</b>	Eclipse Modeling Framework
<b>EBNF</b>	Erweiterten Backus-Naur-Form
<b>ER</b>	Entity-Relationship
<b>EVELIN</b>	Experimentelle Verbesserung des Lernens von Software Engineering
<b>FCC</b>	Fantasy-Challenge-Curiosity

<b>GaML</b>	Gamification Modeling Language
<b>GBL</b>	Game-based Learning
<b>GCDP</b>	Gamified Course Design Process
<b>GCDR</b>	Gamified Course Design Record
<b>GDE</b>	Gamification Design Elemente
<b>GDL</b>	Game Description Language
<b>GEF</b>	Graphical Editor Framework
<b>GMF</b>	Graphical Modeling Framework
<b>GUI</b>	Graphical User Interface
<b>HTML</b>	Hypertext Markup Language
<b>IoC</b>	Inversion of Control
<b>ISR</b>	Information System Research
<b>JDBC</b>	Java Database Connectivity
<b>JiTT</b>	Just-in-Time Teaching
<b>JVM</b>	Java Virtual Machine
<b>KI</b>	Künstliche Intelligenz
<b>LMS</b>	Learning Management System
<b>LOM</b>	Learning Object Metadata
<b>MDA</b>	Model Driven Architecture
<b>MDD</b>	Model Driven Development
<b>MDE</b>	Model Driven Engineering
<b>MDSD</b>	Model Driven Software Design
<b>MOF</b>	Meta Object Facility
<b>MOF<sub>2</sub></b>	Meta Object Facility 2
<b>MOFM<sub>2</sub>T</b>	Meta Object Facility Model to Text
<b>MOOC</b>	Massive Open Online Courses
<b>MTL</b>	Model to Text Language
<b>MVC</b>	Model-View-Controller
<b>OCL</b>	Object Constraint Language
<b>OMG</b>	Object Management Group
<b>QTI</b>	Question & Test Interoperability

<b>RDB</b>	Relationale Datenbank
<b>SCORM</b>	Sharable Content Object Reference Model
<b>SDT</b>	Self-Determination Theory
<b>SLT</b>	Struktur-Lege-Technik
<b>SoC</b>	Separation of Concerns
<b>SQL</b>	Structured Query Language
<b>SRP</b>	Single Responsibility Principle
<b>SysML</b>	Systems Modeling Language
<b>UDL</b>	Universal Design for Learning
<b>UML</b>	Unified Modeling Language
<b>URL</b>	Uniform Resource Locator
<b>VSM</b>	Viewpoint Specification Model
<b>WYSIWYG</b>	What You See Is What You Get
<b>XMI</b>	XML Metadata Interchange
<b>XML</b>	Extensible Markup Language

---

## EINFÜHRUNG

---

### 1.1 PROBLEMKONTEXT UND WISSENSCHAFTLICHER BEITRAG

„[A]ller andern vergessend sitzt der deutsche gelehrte froh über seiner arbeit, dass ihm die augen sich röthen und die knie schlottern; dem student ist dieselbe weise wie angeboren und es bedarf für ihn keines andern antriebs“ (Grimm, 1850, S. 22f.).

Mit diesen Zeilen beschreibt Jacob Grimm 1850 die damalige Lehrsituation in einer seiner Vorlesungen. Während Grimm weit vor den Zeiten der Digitalisierung des Lernens von durchweg intrinsisch motivierten Lernenden ausging, zeigt sich heute, mehr als 150 Jahre später, ein differenziertes und komplexeres Bild in Bezug auf die motivationalen Eigenschaften von Lernenden, nicht zuletzt begründet durch langjährige motivationspsychologische sowie pädagogische Forschung und veränderten sozio-kulturellen Gegebenheiten.

Ein in unterschiedlichen Domänen vielfach eingesetzter Ansatz, der auf einschlägigen motivationspsychologischen Theorien fußt und Lernende individuell zu motivieren vermag, ist *Gamification*. Der Ansatz beschreibt allgemein die Integration von spielerischen Elementen in nicht spielerische Kontexte (Deterding et al., 2011a) und soll für vorliegende Arbeit im Kontext der Hochschullehre betrachtet werden. Während Gamification in der Lehre immer häufiger als Mittel der Motivationssteigerung bei Lernenden angewandt wird (Rughinis, 2013a), existieren wenige Erkenntnisse darüber, wie sich der Einsatz von spielifizierten Elementen auf das Lernen auswirkt (Nah et al., 2014b, S. 407; Morford et al., 2014, S. 36). Vor allem das systematische Design von spielifizierten und digitalen Lernangeboten sowie deren Evaluation zeigen Forschungslücken (Deterding et al., 2013, S. 3263f.). Aus technologischer Perspektive existieren keine geeigneten Werkzeuge, die vor allem Lehrende bei derartigen Designvorhaben unterstützen, wie Dicheva u. Dichev (2016, S. 5f.) und Dichev et al. (2014, S. 81) feststellen.

Die vorliegende Arbeit hat deshalb das Ziel, eine flexible und erweiterbare Systematisierung zu schaffen für die systematische Erstellung von spielifizierten online-Lernangeboten mit Hilfe eines Autorenwerkzeugs, dass auch von nicht-Domänenexperten dafür verwendet werden kann, derartige Lernarrangements effizient zu konzipieren und unmittelbar für Lernende nutzbar zu machen. Ein derartiges Autorenwerkzeug bzw. Autorensystem in Verbindung mit einer wissenschaftlich untermauerten Systematik unterstützt nicht nur die effiziente Erstellung und Implementierung von spielifizierten Lehr-Lernarrangements, sondern trägt ebenso dazu bei, weitere empirische Forschung hinsichtlich der gezielten Wirkungszusammenhänge von Gamification in Lehr-Lernkontexten zu ermöglichen (Dicheva u. Dichev, 2016). Im Zuge vorliegender Forschungsarbeit werden die folgenden Forschungsfragen beantwortet, die als zentrale Desiderate die Arbeit leiten:

1. Welche Elemente von Gamification werden in Lehr-Lernkontexten eingesetzt?
2. Wie können die identifizierten Elemente von Gamification beschrieben werden, bezogen auf Attribute, Zusammenhänge und Regeln?
3. Wie sieht eine aus den empirischen Daten abgeleitete Systematisierung in Form einer grafischen DSML aus?
4. Wie sieht eine Werkzeug-gestützte Umsetzung der erstellten Systematik im Hochschulkontext aus?

Bedingt durch den Gamification-Ansatz und der Anwendung in der Hochschullehre (im Speziellen der Informatik-nahen Ausbildung), handelt es sich bei diesem Forschungsvorhaben um ein interdisziplinäres Vorhaben, welches neben motivationspsychologischen Aspekten, ebenso didaktische Aspekte beinhaltet und bei der Konzeption des Autorensystem berücksichtigt. Unter Einbezug dieser Aspekte liegt das Hauptaugenmerk vorliegender Arbeit jedoch auf der technischen und prototypischen Umsetzung des Systems. Damit wird bewusst davon Abstand genommen, Aussagen über dessen Wirksamkeit auf die Motivation von Lernenden in spielifizierten Lehr-Lernkontexten zu tätigen. Das entwickelte System stellt vielmehr als eine *Enabling Technology* die Basis für weitere Forschungsarbeiten mit derartigen Zielsetzungen dar.

Zusammengefasst behandelt vorliegende Arbeit die folgenden wissenschaftlichen Ziele:

- **Gamification und Lehre:** Beschreibung des Gamification-Ansatzes

und seiner Bestandteile sowie Erfassung des aktuellen Stands der Forschung; Theorieentwicklung und -verifizierung zur strukturellen Vereinbarkeit von kompetenzorientierter Lehre und Gamification; Beschreibung motivationstheoretischer Grundlagen des Gamification-Ansatzes sowie Entwicklung von Designprinzipien zur Förderung von Lernmotivation durch Gamification.

- **Empirische Analyse der Domäne:** Beschreibung einer Dokumentenanalyse samt verwandten Untersuchungen, Einsatz von Methoden, Vorgehen bei der Durchführung und Ergebnisse zur Erfassung von konzeptuellen Anforderungen an die strukturellen Bestandteile von Gamification; Anreicherung und Festigung bzw. Vertiefung der empirischen Ergebnisse aus vorheriger Analyse durch eine Lernplattformanalyse spielifizierter Lernplattformen; Erhebung von konzeptuellen Anforderungen als Exzerpt beider empirischen Analysen.
- **Systematisierung in Form einer Modellierungssprache:** Beschreibung des Domain-specific Modeling-Ansatzes und Zieldefinition für die Umsetzung der konzeptuellen Anforderungen in Form einer domänen-spezifischen Modellierungssprache; Konzeption und Entwicklung der abstrakten Syntax der Modellierungssprache auf Basis der konzeptuellen Anforderungen; Konzeption und Entwicklung der konkreten Syntax der Modellierungssprache unter Zuhilfenahme einer Nutzerstudie mit potentiellen Nutzern
- **Werkzeuge zur Nutzung der Modellierungssprache:** Entwicklung eines Designers, zur Erstellung von Modellen auf Basis der Modellierungssprache; Entwicklung und Anbindung eines Generators zur Transformation von Modellen in Quellcode; Prototypische Entwicklung zweier Zielplattformen in Form eines (u.a. mobilen) Learning Management Systems, in welches jeweils der generierte Quellcode eingesetzt wird und das direkt in die Lehre integriert wird; Beschreibung verwandter Arbeiten.
- **Evaluation:** Beschreibung einer Interview-Feldstudie zur Evaluation der Modellierungssprache und deren Anwendung im dafür vorgesehenen Designer zur Erstellung von spielifizierten Lernarrangements; Abgleich von festgelegten Zielen für die Umsetzung der konzeptuellen Anforderungen mit tatsächlicher Umsetzung; Beschreibung möglicher Einsatzszenarien der entwickelten Werkzeuge, beispielhafte Demonstrati-

on der entwickelten Werkzeuge für die Umsetzung einer spielifizierten Just-in-Time-Teaching-Lerneinheit.

## 1.2 FORSCHUNGSDESIGN

Dieser Arbeit wird als Forschungsansatz der in den Information Systems Research<sup>1</sup> vielbeachtete Beitrag<sup>2</sup> von Hevner et al. (2004) zu Grunde gelegt. In dem darin beschriebenen Design Science Research (DSR)-Ansatz werden zwei konkurrierende und von einander abgrenzbare Paradigmen vereint (Hevner et al., 2004): das empirisch-quantitative (*Behavioural Science*) und das konstruktionsorientierte Paradigma (*Design Science*). Der empirisch-quantitative Teil des Ansatzes zielt darauf ab,

„[...] to develop and justify theories [...] that explain or predict organizational and human phenomena surrounding the analysis, design, implementation, management, and use of information systems“ (Hevner et al., 2004, S. 76).

Aus einer konstruktionsorientierten Sichtweise verfolgt der Ansatz zudem das Ziel, relevante Probleme innerhalb von Organisationen zu lösen und unterstellt dabei einen ingenieur- und artifiziellwissenschaftlichen Charakter (Simon, 1996). Dies geschieht durch die Erstellung von innovativen Artefakten, wie Ideen, Praktiken, technologische Fertigkeiten und Produkten, um die Analyse, die Implementierung, das Management und die Benutzung von Informationssystemen effektiv und effizient zu unterstützen (Hevner et al., 2004; Denning, 1997; Tsichritzis, 1997). Dabei soll ein technologischer Fortschritt erzielt werden, der fundierte empirische Theorien gleichwertig gegenüber stellt und durch diese validiert wird (Hevner u. Chatterjee, 2010, S. 12; Fischer et al., 2010, S. 383; Peffers et al., 2007, S. 49). Die Abbildung 1 verdeutlicht die gegenseitige Wechselwirkung beider Paradigmen im Kontext des Information System Research (ISR).

Aus der untrennbaren Verbindung beider Paradigmen im ISR folgern Hevner et al. Forschungsaktivitäten, welche Sie in einem konzeptuellen Framework (siehe Abbildung 2) strukturieren und für die Positionierung und den Vergleich beider Paradigmen nutzen (Hevner et al., 2004, S. 79). Zelewski unterstellt dem Framework – und damit allen beinhalteten Forschungsaktivitäten – eine Adaptierbarkeit auf alle realwissenschaftlichen Disziplinen, welche

<sup>1</sup> Entsprechend dem im deutschen Sprachraum bekannten Begriff der Wirtschaftsinformatik (Zelewski, 2007, S. 71).

<sup>2</sup> Eine Übersicht über Beiträge, die diesen Forschungsansatz nutzen, findet sich beispielsweise in Zelewski (2007, S. 72).



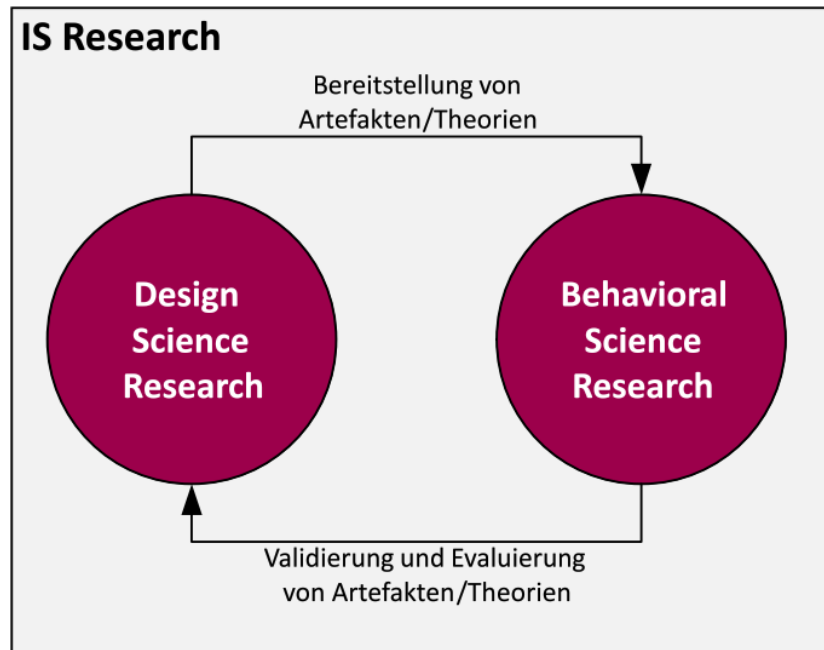


Abbildung 1: Komplementarität von Design Science und Behavioral Science nach Hevner u. Chatterjee (2010, S. 11)

das Zusammenspiel von Menschen und Technik beleuchten (Zelewski, 2007, S. 73).

Das Framework besteht aus drei Säulen, welche sich in Beziehung zueinander setzen lassen. Die Realweltumgebung besitzt als Determinanten die Menschen, die Organisationen sowie die Technik (Hevner et al., 2004, S. 79). Alle drei Determinanten bilden Praxisanforderungen aus, welche in Form von Zielen, Aufgaben, Problemen und Möglichkeiten auftauchen können (Hevner et al., 2004, S. 79). Die Menschen innerhalb einer Organisation nehmen diese Praxisanforderungen wahr und erfüllen dabei definierte Rollen, besitzen unterschiedliche Fähigkeiten und Eigenschaften. Sie sind eingegliedert in eine Organisation, die strategische Vorgehensweisen besitzt, sich dabei Strukturen bedient, die abhängig von einer Organisationskultur sind, sowie Geschäftsprozesse nutzt (Hevner et al., 2004, S. 79). Zudem werden Praxisanforderungen gegenüber der Determinante Technik positioniert und bewertet. Dies geschieht durch eine relative Gegenüberstellung von Praxisanforderungen mit den (vorhandenen) Bestandteilen Infrastruktur, Anwendungen, Kommunikationsfähigkeit sowie Entwicklungsfähigkeit. Aus Sicht des Forschers erzeugen die genannten Determinanten der Realweltumgebung relevante Praxisanforderungen (Hevner et al., 2004, S. 79; Hevner u. Chatterjee, 2010, S. 17).

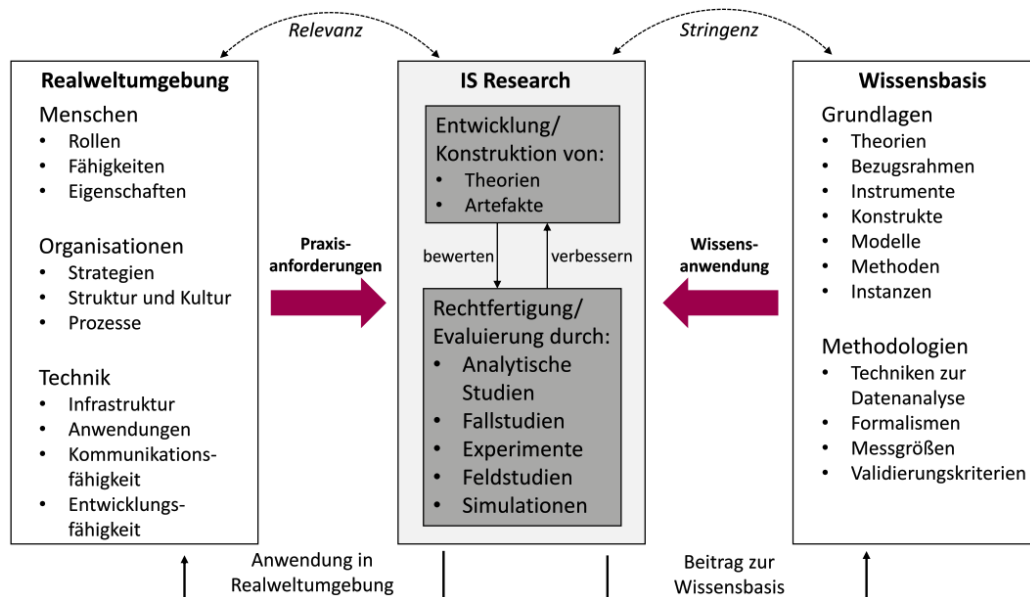


Abbildung 2: Information System Research Framework, eigene Übersetzung nach Hevner et al. (2004, S. 80)

Eine weitere Säule des Frameworks ist die Wissensbasis. Sie besteht aus den Determinanten Grundlagen sowie Methodologien und stellt Wissensmaterial bereit, durch das ISR von erkenntnistheoretischer Seite durch Theorien, Frameworks, Instrumente, Konstrukte, Modelle, Methoden oder Instanzen unterstützt wird. Damit knüpft sie an die Disziplin des Wissensmanagements an, welche eine informationswissenschaftliche Basis für ISR bereitstellt (Zelowski, 2007, S. 75; Hevner et al., 2004, S. 80). Methodologien wie Techniken zur Datenanalyse, Formalismen, Messgrößen oder Validierungskriterien stellen Richtlinien zur Verfügung, welche im ISR, der dritten und zentralen Säule des Frameworks nach Hevner et al., ihre Anwendung finden. Aus einer geeigneten Anwendung von Grundlagen und Methodologien im ISR wird diese dritte Säule durch eine Stringenz unterstützt, welche die IS Forschung in einen wissenschaftlichen Kontext stellt.

In der Säule des ISR fließen die Realweltumgebung und Wissensbasis bei der Entwicklung und Bewertung von neuen Artefakten und Theorien zusammen (siehe Abbildung 1). Aus Prozesssicht ließe sich zwischen den Determinanten Entwicklung und Konstruktion von Artefakten und Theorien und deren Rechtfertigung oder Evaluierung durch Methodologien ein iterativ-inkrementelles Vorgehen annehmen. Dabei ist jedes Inkrement das Ergebnis einer Iteration und somit eine neue und (idealerweise) verbesserte Version eines Artefakts oder einer Theorie, welche auf einer vorhergehenden Bewertung basiert (Hevner u. Chatterjee, 2010, S. 18f.). Hevner u. Chatterjee sprechen in diesem Zusammenhang von einem *Design Cycle* (Designkreislauf), der so lange

iteriert, bis das Artefakt eine gewisse Reife im Sinne der zu Beginn anvisierten Eignung erreicht hat (Hevner u. Chatterjee, 2010, S. 18). Hierzu werden Anforderungen (und deren Messbarkeit) zu Beginn der Forschungsaktivitäten definiert, anhand deren eine Beurteilung des Artefakts oder der Theorie stattfindet. Derartige Anforderungen sind Teil des *Relevance Cycles* (Relevanzkreislauf), und äußern sich durch - für die Realweltumgebung relevante - Praxisanforderungen (Hevner u. Chatterjee, 2010, S. 17). Der *Rigor Cycle* (Stringenzkreislauf) sieht eine Anwendung der bereits oben beschriebenen Determinanten der Säule Wissensbasis vor. Dem fügen die Autoren hinzu, dass das Einbeziehen von Erfahrungen und Expertise zu aktuellen Anwendungen ebenso notwendig ist wie das Betrachten von Meta-Artefakten<sup>3</sup> aus der entsprechenden Zieldomäne (Hevner u. Chatterjee, 2010, S. 18). Letztere können auch aus anderen, angrenzenden Domänen stammen. Allerdings soll für die zu erstellenden Artefakte innerhalb des Designkreislaufes nicht der Anspruch gelten, dass „all design decisions and design processes be based on grounded behaviour or mathematical theories“ (Hevner u. Chatterjee, 2010, S. 18), da dies sonst zu einer Hemmung in der Kreativität bei den Forschenden führen könnte. Nach Zelewski deuten Hevner et al. damit an, dass es „nicht um eine 'gleichberechtigte' Behandlung von konstruktionsorientiertem und empirisch-quantitativen Paradigma geht, sondern um eine Aufwertung des konstruktionsorientierten Paradigmas“ (Zelewski, 2007, S. 77).

Das Ergebnis des ISR kommt in der Realweltumgebung zur Anwendung und wird dort bewertet (Hevner u. Chatterjee, 2010, S. 17). Die Bewertung erfolgt auf Basis von messbaren Verbesserungen<sup>4</sup>, die das entwickelte Ergebnis des ISR im Einsatz zeigt. Auch in Bezug auf die Wissensbasis kommt es zu einem Rückfluss in Form von den erzeugten Artefakten oder Theorien, der Erweiterung bestehender Grundlagen oder Methodologien, sowie der Erfahrung durch den Erfahrungsgewinn während der IS-Forschung an sich (Hevner et al., 2004, S. 81; Hevner u. Chatterjee, 2010, S. 18). Fischer et al. unterstellen dieser Ansicht von Hevner et al., dass ein additiver Beitrag eines Artefakts oder einer Theorie zu dem Korpus der Wissensbasis nur erfolgen kann, wenn ein „gewisser Grad an Allgemeingültigkeit“ gegeben ist (Fischer et al., 2010, S. 385).

<sup>3</sup> Gemeint sind bereits existierende Artefakte oder Prozesse, die von Relevanz für das IS Forschungsvorhaben sein können (Iivari, 2007, S. 47).

<sup>4</sup> Zum Beispiel wie von Rossi oder Järvinen vorgeschlagen, durch den Einsatz von *Action Research* (Rossi, 2009; Järvinen, 2007), obwohl diese Anwendung auf Grund der Unterschiedlichkeit beider Ansätze durchaus strittig ist (Iivari, 2007, S. 53).

### 1.3 ORGANISATORISCHER RAHMEN

Diese Forschungsarbeit ist im Rahmen des durch das Bundesministerium für Bildung und Forschung (BMBF) geförderte Projekt Experimentelle Verbesserung des Lernens von Software Engineering (EVELIN) (Abke et al., 2012a; Abke et al., 2012b) angefertigt worden. Das Projekt EVELIN ist ein Verbundprojekt, an dem sechs bayerische Hochschulen beteiligt sind, welche sich mit dem Thema beschäftigen, wie die Lehre von Software Engineering an Hochschulen praktisch fundiert verbessert werden kann. Im Rahmen des fünfjährigen Projekts der ersten Förderphase werden nicht nur neue Lehr- bzw. Lernarrangements eingesetzt und anschließend evaluiert, sondern auch bisher nur in Ansätzen vorhandene Forschung zum Thema Kompetenzentwicklung und -förderung im Software Engineering betrieben. Das Projekt wurde 2017 um weitere vier Jahre verlängert und befindet sich aktuell in der zweiten Förderphase.

### 1.4 STRUKTUR DER ARBEIT

**KAPITEL 1: EINFÜHRUNG** Das Einführungskapitel gibt einen Überblick über den Kontext der vorliegenden Arbeit und zeigt die Forschungsdesiderate, welche diese Arbeit behandelt.

**KAPITEL 2: GAMIFICATION – DIE ANATOMIE DES ANSATZES** Das zweite Kapitel behandelt aktuelles Hintergrundwissen zu Gamification und beschreibt den Ansatz aus unterschiedlichen Sichtweisen. Es definiert relevante Terminologie, grenzt den Ansatz von verwandten Ansätzen ab und beschreibt seine Bestandteile. Zudem entwickelt es eine Theorie zur Vereinbarkeit von Gamification und kompetenzorientierter Lehre und validiert diese durch eine Feedback-Studie.

**KAPITEL 3: MOTIVATIONSFÖRDERUNG DURCH GAMIFICATION** Das dritte Kapitel nimmt eine motivationstheoretische Perspektive auf Gamification ein und präsentiert dafür relevantes Hintergrundwissen aus dem Bereich der Motivationspsychologie. Dieses Wissen dient nicht nur der Erklärung auf welchen motivationalen Aspekten Gamification basiert, sondern bildet zudem die Grundlage für die Entwicklung von Designheuristiken für Gamification, mit denen die Lernmotivation von Lernenden gesteigert werden kann. Die demonstrierten Designheuristiken werden an unterschiedlicher Stelle der Ar-

beit direkt oder indirekt wiederverwendet, wie beispielsweise in den Kapiteln 5, 7, 8.

**KAPITEL 4: EMPIRISCHE STUDIEN ZU GAMIFICATION IN DER LEHRE**  
Das vierte Kapitel zeigt zwei Analysen, die den aktuellen Forschungs- und Praxisstand von Gamification in Lehr-Lernkontexten systematisch erfassen, um daraus für das Design von spielifizierten Lehr-Lernarrangements relevante Domänenkonzepte zu exzerpieren. Hierfür wurde eine Dokumentenanalyse und darauf aufbauend eine Analyse von bestehenden Lernplattformen durchgeführt. Erstere fokussiert die Literatur, während zweitens die Perspektive aus der Praxis erfasst. Abschließend werden die beiden ersten Forschungsfragen beantwortet, welche diese Arbeit leiten.

**KAPITEL 5: EMENDO DSML** Das fünfte Kapitel widmet sich der systematischen Erstellung der *Emendo DSML*, einer domänenspezifischen Modellierungssprache. Es stellt Hintergrundwissen für den *Domain-specific Modeling*-Ansatz bereit und zeigt, wie die aus beiden vorherigen Analysen exzerpierten Domänenkonzepte sukzessive in eine abstrakte und empirisch gestützte konkrete Syntax der Emendo DSML überführt werden. In diesem Kapitel wird zudem die dritte Forschungsfrage beantwortet.

**KAPITEL 6: EMENDO DESIGNER UND GENERATOR** Das sechste Kapitel beschreibt den *Emendo Designer*, der als Modellierungswerkzeug fungiert und dabei die Emendo DSML integriert. Es geht auf relevante Aspekte bei der Modellierung mit dem Designer und der DSML ein und zeigt zudem Details zum *Emendo Generator*, der auf der DSML basierende Modelle in Quellcode transformiert. Die Integration des Generators wird ebenso thematisiert, wie zu Emendo existierende verwandte Arbeiten.

**KAPITEL 7: EMENDO LEARNING MANAGEMENT SYSTEM** Im siebten Kapitel liegt das Hauptaugenmerk auf der Entwicklung des *Emendo Learning Management Systems*. Es zeigt zunächst eine systematische Anforderungsanalyse und beschreibt daraufhin die Systemarchitektur samt relevanter Komponenten unter Verwendung von verschiedenen Diagrammtypen der UML. Die GUI und Benutzerinteraktionsmöglichkeiten werden ebenso thematisiert. Hinzukommend wird durch das *Mobile Emendo LMS* die Portabilität des Emendo LMS demonstriert.

**KAPITEL 8: EVALUATION** Das achte Kapitel beschreibt eine qualitativ angelegte Feldstudie, mit dem Ziel, das allgemeine Konzept von Emendo sowie die Anwendung des Emendo Designers und der darin eingebetteten Emendo DSML durch Lehrende zu beurteilen. Außerdem werden in Kapitel fünf erhobene generischen Anforderungen an die Emendo DSML mit der tatsächlichen Umsetzung abgeglichen. Abschließend erfolgt eine beispielhafte Demonstration, wie Emendo zur Konzeption einer spielifizierten *JiTT-Lerneinheit* verwendet werden kann.

**KAPITEL 9: ZUSAMMENFASSUNG UND AUSBLICK** Das letzte Kapitel fasst diese Arbeit kapitelweise zusammen und stellt jeweils wesentliche wissenschaftliche Beiträge heraus. Darüber hinaus wird ein Ausblick über zukünftige und an diese Arbeit anknüpfende Forschungsarbeiten gegeben.

---

## GAMIFICATION - DIE ANATOMIE DES ANSATZES

---

*„I think there’s a good case to be made  
that having fun is a key evolutionary advantage  
right next to opposable thumbs in terms of importance.  
Without that little chemical twist in our brains  
that makes us enjoy learning new things,  
we might be more like the sharks and ants of the world.“*

— Raph Koster (Koster, 2013, S. 96)

### 2.1 HINFÜHRUNG

Gemäß dem DSR Forschungsdesign nach Hevner et al. soll in einem ersten Schritt der Fokus auf die Wissensbasis gelegt werden. Hierzu wird in diesem Kapitel ein grundlegendes und für die weitere Arbeit notwendiges Verständnis über das allgemeine Konzept von Gamification aufgebaut und von verwandten Konzepten abgegrenzt. Des weiteren werden etablierte Modelle zur Strukturierung von Game Design-Elementen bzw. Gamification Design-Elementen präsentiert, die für die Analyse von bereits bestehenden oder noch zu erstellenden spielifizierten Anwendungen hilfreich sind.

Zunächst soll eine Annäherung über den Begriffsstamm von Gamification erfolgen und der Fokus auf der Definition des Begriffs *Game* liegen. Zieht man beispielsweise die Definition eines Games von Huizinga als einen der wichtigsten Theoretiker des Spiels heran, stellt man fest, dass diese grundlegende Determinanten zur Charakterisierung eines Games beinhaltet. Er formuliert:

„Der Form nach betrachtet, kann man das Spiel also zusammenfassend eine freie Handlung nennen, die als *nicht so gemeint* und außerhalb des gewöhnlichen Lebens stehend empfunden wird und trotzdem den Spieler völlig in Beschlag nehmen kann, an die kein materielles Interesse geknüpft ist und mit der kein Nutzen erworben

wird, die sich innerhalb einer eigens bestimmten Zeit und eines eigens bestimmten Raums vollzieht, die nach bestimmten Regeln ordnungsgemäß verläuft und Gemeinschaftsverbände ins Leben ruft, die ihrerseits sich gern mit einem Geheimnis umgeben oder durch Verkleidung als anders von der gewöhnlichen Welt abheben“ (Huizinga, 2013, S. 22).

Fasst man seine Definition zusammen, extrahiert die wesentlichen Bestandteile und reichert sie mit der von dem Videospieltheoretiker Juul durchgeführten Untersuchung von weiteren Game-Definitionen<sup>1</sup> an, so kommt man zu sechs Charakteristika, durch die sich Games auszeichnen und damit definieren lassen (Juul, 2005, S. 36):

Ein Game...

- ... ist ein rollenbasiertes System,
- mit einem variablen und quantifizierbaren Ergebnis,
- bei dem unterschiedliche quantifizierbare Ergebnisse unterschiedlichen Werten (positiv und negativ) zugeordnet sind,
- der Spieler Anstrengungen aufbringen muss (Spiele sind herausfordernd), um das Ergebnis zu beeinflussen,
- sich der Spieler emotional mit dem Ergebnis verbunden fühlt (es also eine gewisse Relevanz besitzt)
- und die Konsequenzen der Spielaktivität (im realen Leben) optional und verhandelbar sind.

Bei diesen Charakteristiken ist anzumerken, dass diese zwar durchaus weit verbreitet sind, sich aber die Forschungsgemeinde in dieser Domäne bisher nicht auf eine spezifische und allgemeingültige Definition eines Games einigen konnte, so Klabbers (Klabbers, 2008). Vielmehr seien die existierenden Definitionen dieses Begriffs begründet durch biografische sowie soziale Einflüsse, individuelle Erfahrungen und heterogene Ziele (Greenblat, 1974, S. 5). Diese Uneinigkeit in der Definitionsfrage ist vor allem unter dem Gesichtspunkt bemerkenswert, dass Aarseth – sieben Jahre vor der Attestierung durch Klabbers – das Jahr 2001 als das „Year One“ bezeichnet, also das Geburtsjahr

<sup>1</sup> Juul integrierte auch unter anderem in seine Untersuchung die von Salen u. Zimmerman häufig rezitierte und bekannte Definition von einem Game: „A game is a system in which players engage in artificial conflict, defined by rules, that results in a quantifiable outcome“ (Salen u. Zimmerman, 2004, S. 80).



von Studien zu Computer Games (Aarseth, 2001). Aarseth gibt an, dass im Vorfeld zwar Studien in diesem Bereich durchgeführt und dokumentiert wurden, sich diese Disziplin jedoch erst 2001 als eine international aufstrebende und zukunftsfähige Forschungsdisziplin manifestieren konnte (Aarseth, 2001).

Im Folgenden soll der Fokus darauf liegen, zu einer Definition von Gamification zu kommen, um den Begriff fortan von weiteren angrenzenden Domänen oder synonym verwendeten Begrifflichkeiten abzugrenzen. Dabei sind einige, das Game Design betreffende Teile der oben genannten allgemeinen Definition von Juul im Gamification-Ansatz in verwandten Ausprägungen wieder zu finden.

## 2.2 DEFINITORISCHER RAHMEN

Gamification<sup>2</sup> wird nach Deterding et al. (2011a) definiert als „the use of game design elements in non-game contexts“ (Deterding et al., 2011a, S. 10). Deterding et al., die Autoren dieser weit verbreiteten Begriffsdefinition, schafften damit eine generische Basis, aus der sich viele weitere Definitionen kontext- bzw. domänenspezifisch ableiten lassen. So findet sich stark angelehnt an die Definition von Deterding et al. die Definition von Werbach u. Hunter. Diese beschreiben Gamification als „the use of game elements and game-design techniques in non-game contexts“ (Werbach u. Hunter, 2012, S. 26). Die Erweiterung um den Begriff der „game-design techniques“ deutet darauf hin, sich bei einem Einsatz von spielerischen Elementen bestehender und erprobter Standards aus dem Bereich von Games zu bedienen und nicht davon auszugehen ist, dass die bloße Existenz eines spielerischen Elements aktivierend und motivierend für einen Nutzer wirkt (Huotari u. Hamari, 2012, S. 19; Werbach u. Hunter, 2012, S. 60). In einer weiteren Definition aus der Domäne des Marketings finden sich explizite und damit teilweise eingeschränkte Zielsetzungen von Gamification, die als fester Bestandteil des Begriffsumfangs gesehen werden. So integrieren Zichermann u. Cunningham zudem das Wirken auf den Faktor Mensch, indem sie Gamification beschreiben als den Prozess „[...] of game-thinking and game mechanics to engage users and solve problems“ (Zichermann u. Cunningham, 2011, S. XIV). Eine weitere in der Marketingdomäne verortete Definition stammt von Huotari u. Hamari. Basierend auf untersuchter Literatur zu Games und Service Marketing definieren sie Gamification als

<sup>2</sup> In deutschsprachiger Literatur wird häufig der Begriff der Spielifizierung als Synonym verwendet (siehe z. B. Stampfl 2012).

„[...] a form of service packaging where a core service is enhanced by a rules-based service system that provides feedback and interaction mechanisms to the user with an aim to facilitate and support the users' overall value creation“ (Huotari u. Hamari, 2011, S. 3).

Sie grenzen sich dabei von der Definition von Deterding et al. vor allem dadurch ab, dass im als Kern der Definition ein rollenbasiertes Servicesystem steht, welches Feedback und Interaktionsmechanismen für Nutzer bereitstellt, jedoch gleichzeitig mehr als nur ein Game oder spielifizierte Service darstellt (Deterding, 2011b, S. 13). Eine derartige Beschreibung, so Deterding, wäre anwendbar auf „[...] almost any interactive system“ (Deterding, 2011b, S. 13) und besitzt damit einen holistischen Charakter, welcher kurze Zeit später von Huotari u. Hamari weiter eingeschränkt wird: „A process of enhancing a service with affordances for gameful experiences in order to support user's overall value creation“ (Huotari u. Hamari, 2012, S. 19). Bei beiden Definitionen von Huotari u. Hamari ist jedoch ersichtlich, dass sie sich von dem Begriff der Game Design-Elemente distanzieren. Dies begründen die Autoren unter anderem dadurch, dass für sie kein definiertes und abgeschlossenes Set an Game Design-Elementen existiert, durch welches Games oder Gamification als Teilmenge definierbar wären und zudem der bloße Einsatz solcher Elemente nicht automatisch ein Spielempfinden (Game Experience) beim Nutzer hervorrufen (Huotari u. Hamari, 2012, S. 18f.).

Eine ähnliche Einschränkung nimmt Paharia in seiner Definition vor. Aus seiner Perspektive geht es bei dem Ansatz um „[...] motivating people through data“ (Paharia, 2014, S. 1). Der Datenbegriff impliziert eine materialisierte und nach syntaktischen Regeln dargestellte Aufbereitung von Wissen unter Nutzung eines Datenträgers (Kuhlen, 1989, S. 14). Dabei ist nicht gesagt, wie genau diese Aufbereitung zu erfolgen hat, jedoch wird von Paharia vorausgesetzt, dass diese zu erfolgen hat, um Menschen motivieren zu können und somit einen gewünschten Verhaltenseffekt zu erzielen. Die Frage nach der Art der Aufbereitung für den entsprechenden Adressaten bzw. Konsumenten der Informationen mag eine individuelle Frage sein und nicht pauschal beantwortbar. Dennoch kann auf Erkenntnisgrundlage der Motivationsforschung angenommen werden, dass sich die Motive des Handelns bei Individuen durch gezielte Anstöße beeinflussen lassen, um beispielsweise persönliche Ziele zu erreichen und damit einen Output zu erzeugen, der ohne diesen Trigger unter Umständen nicht möglich gewesen wäre (Deci u. Ryan, 1985; Dresel u. Lämmle, 2011; Müller, 2014; Smolka, 2004).

Dass die Verhaltensänderung von Personen in Bezug auf ihre Einstellungen zu Sachverhalten einen der Kernpunkte von Gamification bildet, stellt

auch Burke fest, indem er das Konzept beschreibt als „[...] the use of game mechanics and experience design to digitally engage and motivate people to achieve their goals“ (Burke, 2014b, S. 1). Jedoch erfolgt hier eine weitere Fokussierung, die von anderen Forschern konträr gesehen wird. Der Ansatz hat nicht nur in der digitalen Welt seine Einsatzberechtigung. So sprechen sich beispielhaft Kapp, Danelli und Groh gleichermaßen für einen Einsatz außerhalb der digitalen Welt aus, beispielsweise eingebettet in Lernaktivitäten, die direkt in Lehrveranstaltungen stattfinden (Danelli, 2015; Groh, 2012; Kapp, 2014). Laut Kapp sollte eher das Design<sup>3</sup> von Gamification vordergründig betrachtet werden, als sich auf Technologieaspekte zu fokussieren, denn diese fungieren in erster Linie als ein Vehikel (Kapp, 2014). Seine Definition „[...] Gamification is using game-based mechanics, aesthetics and game thinking to engage people, motivate action, promote learning, and solve problems“ (Kapp, 2012a, S. 10) spezifiziert stärker die zu berücksichtigenden Bestandteile des Begriffsverständnisses und nennt die Ziele, die mit Hilfe von Gamification in Lehr-Lernkontexten erreicht werden sollen.

Damit werden die ursprünglich von Deterding et al. betitelten *game design elements* von Kapp analogisiert, als die Fokussierung auf *game-based mechanics*, *aesthetics* und *game thinking*, was der Empfehlung von Huotari u. Hamari und Werbach u. Hunter folgt, sich bestehender Techniken und Praktiken aus dem Game Design Bereich zu bedienen (Huotari u. Hamari, 2012; (Werbach u. Hunter, 2012)). Die von Deterding et al. angesprochenen *non-game contexts* sind hier für den Bereich Lehr-Lernkontexte instanziiert und damit die Zielsetzungen *motivate action*, *promote learning* und *solve problems* auf die Lernenden bezogen.

### 2.2.1 Begriffsheterogenität

Wie man erkennt, herrscht in der Wissenschaft kein einheitliches und einschlägiges Verständnis darüber, wie mit dem Begriff Gamification allgemeingültig umgegangen werden soll (Deterding, 2016, S. 104; Seaborn u. Fels, 2015; Bartel u. Hagel, 2016b). Je nach Kontext wird der Begriff definiert durch die eingesetzten spielerischen Elemente, die erwünschten Zielsetzungen bzw. Outcomes, die Nähe zu verwandten Fachgebieten wie den Serious Games oder die Existenz von einem spezifischen Kontext (Deterding, 2016, S. 104). Dies führte in der Vergangenheit zu einer Begriffsheterogenität, die in der entspre-

<sup>3</sup> Kapp prägt hierzu den Begriff der „design sensibility“ (Kapp, 2014). Dieser bedarfsgetriebene Designgedanke (im Gegensatz zum technologiebasierten Vorgehen) wird auch von weiteren Forschern, wie beispielsweise Dichev u. Dicheva geteilt (Dichev u. Dicheva, 2017).

chenden Fachliteratur häufig thematisiert wird und schließlich die unpräzise Verwendung des Begriffs in einem „[...] informal umbrella term for the use of video game elements [...]“ (Deterding et al., 2011b, S. 2425) mündete. Nach Seaborn u. Fels wird der Begriff Gamification zudem in der Literatur dafür genutzt, weitere Konzepte über die eigentliche Definition des Begriffs hinaus zu beschreiben<sup>4</sup> (Seaborn u. Fels, 2015, S. 8), was die These einer Begriffsverwendung als „informal umbrella term“ weiter unterstützt. Dabei handelt es sich zum einen um das Erstellen oder die Nutzung eines Games innerhalb eines nicht Medien-basierten Kontextes oder Ziels und zum anderen um die Umwandlung eines bestehenden Systems in ein Game (Seaborn u. Fels, 2015, S. 8). Die Autoren identifizieren hierzu die häufig in der Literatur verwendeten Aktionswörter, wie zum Beispiel *gamified* oder *gamify*, und leiten daraus ab, dass in derartigen Arbeiten eine Anwendung des Gamification-Konzepts im Vordergrund steht, im Gegensatz zu einer theoretischen oder konzeptuellen Betrachtung des Ansatzes. Die Autoren schlagen als Konsequenz des fehlenden Konsens des Gamification-Begriffs eine Konsens-Definition vor, welche auf den Definitionen von Deterding et al. (2011a), Huotari u. Hamari (2012), Werbach u. Hunter (2012) und Zichermann u. Cunningham (2011) basiert und definieren Gamification als „the intentional use of game elements for a gameful experience of non-game tasks and contexts“ (Seaborn u. Fels, 2015, S. 8).

### 2.2.2 Definition für diese Forschungsarbeit

Die Folge eines derartig weitgefassten Begriffsverständnisses ist eine ebenso weitgefasste Anwendung des Konzepts und, damit einhergehend, die Notwendigkeit einer zu Grunde liegenden Definition für diese Arbeit zunächst unabdingbar. Für diese Definition sei eine wichtige Erkenntnis aus der pädagogischen Psychologie zu Grund gelegt:

„More recent research on behavior modification has suggested that behaviors can be controlled by adjusting the antecedents or the consequences on an action. That is controlling the possibilities for a person to take a certain action or controlling the feedback of an action“ (Bang et al., 2009, S. 2).

Da Gamification für diese Arbeit im Lichte eines Lehr-Lernkontexts steht, spricht nichts gegen eine Adaption des allgemeinen Begriffsverständnisses

<sup>4</sup> Beispielhaft kann hier das Begriffsverständnis von Rughinis angeführt werden, welcher Gamification in Lernkontexten nur dann als das Konzept ansieht, wenn das resultierende spielifizierte System durch einen Lernenden spielbar ist und damit ein simples Gameplay existiert (Rughinis, 2013b, S. 524).

auf die spezifische Domäne. Für diese Arbeit soll deshalb die folgende Definition zu Grunde gelegt werden:

Gamification ist ein Konzept, welches spielerische Elemente und Prozesse in Lernaktivitäten integriert, um die Lernmotivation zu erhöhen und das Verhalten der Lernenden dadurch nachhaltig ändert.

Erkennbar darin sind wesentliche Bestandteile, die den Begriff inhaltlich prägen, sowie dem Konzept ein eindeutiges Ziel vorschreiben. Bei den einzelnen Bestandteilen handelt es sich im Konkreten um die Folgenden:

- Spielerische Elemente.
- Spielerische Prozesse.
- Lernaktivitäten.
- Lernmotivation.
- Nachhaltige Änderung des Verhaltens von Lernenden.

Bevor die Anatomie dieses Ansatzes diskutiert wird, soll zunächst eine Verortung des Begriffs gegenüber angrenzenden Forschungsdomänen stattfinden. Auch dabei lässt sich die nicht durchweg vorhandene Trennschärfe erkennen, welche bedingt durch ein uneinheitliches Begriffsverständnis auftritt.

### 2.3 VERORTUNG DES GAMIFICATION-BEGRIFFS

Für eine Begriffsverortung häufig herangezogene Darstellung ist die Matrix (siehe Abbildung 3) von Deterding et al. (Deterding, 2016, S. 105). Sie unterscheidet zwei Dimensionen mit jeweils zwei Polen. Auf der vertikalen Achse ist die von Caillois<sup>5</sup> getroffene Festlegung visualisiert, welcher in grundsätzlich zwei Formen des menschlichen Spielens unterscheidet und sich dies innerhalb eines Kontinuums, begrenzt durch die zwei Pole der Achse, bewegt (Caillois, 2001, S. 13): Der erste Pol wird benannt als *paidia*, das Prinzip des „diversion, turbulence, free improvisation, and carefree gaiety“ (Caillois, 2001, S. 13) spielens, was sich prototypisch im kindlichen Spiel zeigt (Deterding, 2016, S. 104). Dem gegenübergestellt steht *ludus*, als ein Spielen, welches charakterisiert wird durch „arbitrary, imperative, and purposively tedious conventions“ (Caillois, 2001, S. 13) und damit dem Spieler vergleichsweise mehr

<sup>5</sup> Die Arbeit von Caillois wird in der Forschungscommunity als eine Antwort bzw. Weiterentwicklung der Werke von Johan Huizinga, speziell *Homo Ludens* Huizinga (2013) gesehen (siehe z. B. (Henricks, 2010)).

„effort, patience, skill, or ingenuity“ (Caillois, 2001, S. 13) abverlangt, als es eine paidiasche Spielweise tut.

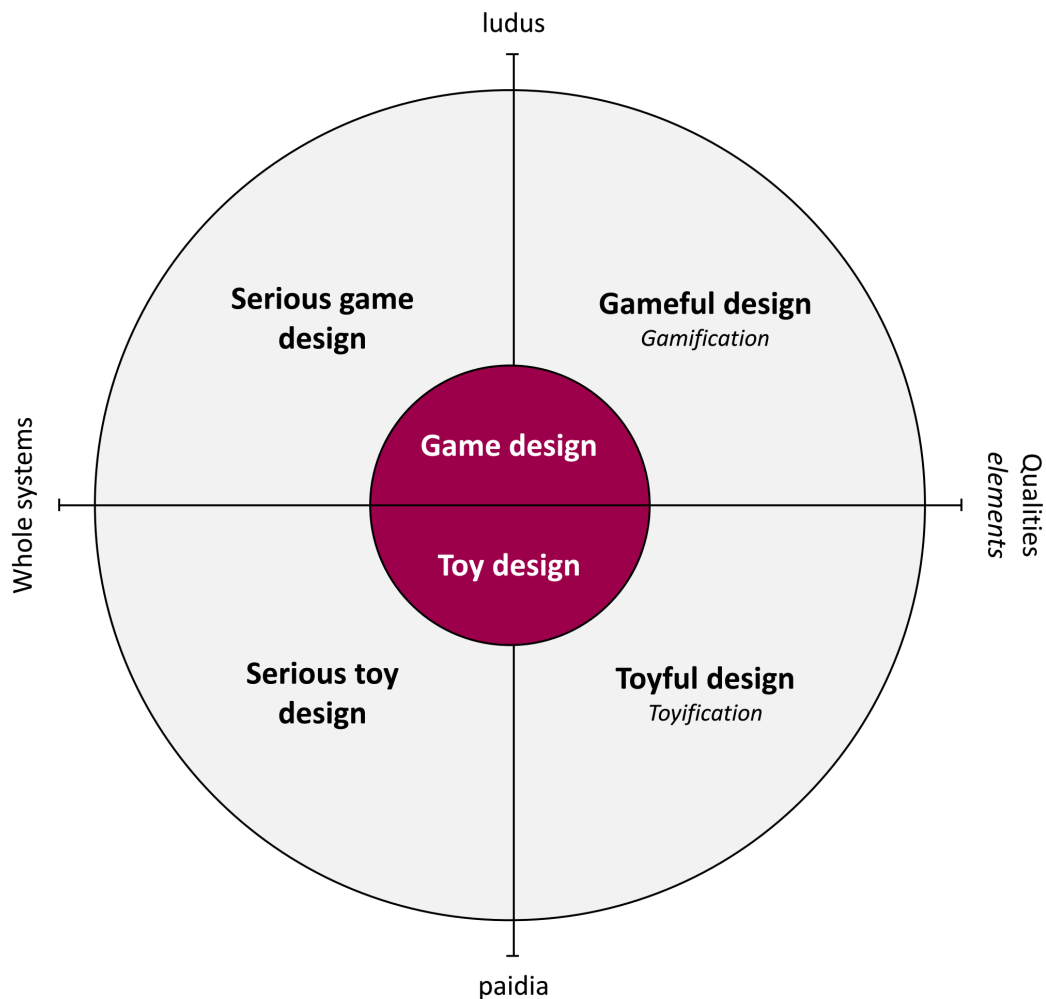


Abbildung 3: Einordnung von Gamification in die angrenzende Begriffswelt (vgl. Deterding (2016, S. 105))

Bei der horizontalen Achse unterscheiden Deterding et al. bezogen auf den Designgegenstand in ganze Systeme (Full-fledged Games (Deterding et al., 2011a, S. 13)) und einzelne Elemente (Qualities) (Deterding, 2016, S. 105)). Durch diese Unterteilung ergeben sich vier voneinander abgrenzbare Quadranten (siehe Abbildung 3).

### 2.3.1 Serious Game Design

Deterding et al. verstehen unter *Serious Game Design* die Erstellung von *ludic wholes*, also das Design oder Deployment von vollständigen Spielen für ernsthafte bzw. zielgerichtete Zwecke (Deterding, 2016, S. 105).

Folgt man der Definition von Göbel et al., so handelt es sich bei Serious Games, um „[...] Spiele mit einem übergeordneten Zweck abgesehen der reinen Unterhaltung“ (Göbel et al., 2014, S. 547). Diese ist erkennbar angelehnt an die Interpretation von Michael u. Chen, welche unter einem Serious Game, ein Game verstehen, „[...] in which education (in its various forms) is the primary goal, rather than entertainment“ (Michael u. Chen, 2005, S. 17).

Spielerische Anwendungen besitzen ein hohes Maß an Dualität bezogen auf ihre Begrifflichkeit, was der ebenfalls nicht vorhandenen Trennschärfe von spielifizierten Anwendungen (Gamification) und vollständigen Serious Games geschuldet ist (Deterding et al., 2011a; Dormans, 2012). Deterding et al., Richter et al. sowie Landers fügen zudem hinzu, dass die Grenzen zwischen beiden Bereichen fließend sein können (Landers, 2014, S. 4; Richter et al., 2015, S. 23, Deterding et al., 2011a). Begründet sei dies unter anderem dadurch, dass sowohl Gamification als auch Serious Gaming eine gemeinsame Game Design-Element-Taxonomie zu Grunde liegt (Werbach u. Hunter, 2015, S. 3). Die von Deterding et al. vorgeschlagene Unterscheidung, wird von Bedwell et al. weiter detailliert. Sie sprechen von *Game Attributes* (beispielsweise Kontrolle, Immersion oder Regeln bzw. Zielen) und verstehen darunter, die Bestandteile einer *Game Design-Element-Taxonomie*. Während Game Attributes in allen Serious Games vorkommen und sich anhand deren Ausmaß (im Sinne von Häufigkeit) sowie der Art und Weise, wie sie für einen Benutzer erfahrbar werden, differenzieren lassen, so werden Game Attributes im Kontext von Gamification nur vereinzelt extrahiert und in Lehr-Lernkontexte adaptiert (Bedwell et al., 2012, S. 752f.; Landers, 2014, S. 6).

Zudem sollen Game-Technologien und spielerische Ansätze genutzt werden, die Kombinationen mit Technologien und (wissenschaftlich-technischen) Ansätzen und Konzepten ergeben und schließlich ein breites Anwendungsspektrum besitzen (Göbel et al., 2014). Es lässt sich schlussfolgern, dass Serious Games als eine Instanz von Games betrachtet werden können, die einem Lernzweck dienlich sind und nicht nur in erster Linie den Spieler unterhalten sollen (Michael u. Chen, 2005, S. 17; Werbach u. Hunter, 2012, S. 33).

**Beispiel:** Bei dem Serious Game *SimSE* Navarro (2006) lernen Studierende einschlägige Prozesse aus dem Software Engineering kennen und partizipieren rollenabhängig an der jeweiligen simulationsbasierten Durchführung. Die Abbildung 4 zeigt einen Screenshot aus dem grafischen User Interface, welches durch Lernende bedient wird. Darauf zu sehen ist ein typisches Büroumfeld (mitte), teilnehmende Spieler (mitte und rechts) bzw. deren Rollen sowie eine Übersicht über aktuelle Aktivitäten (rechts), Artefakte (oben) oder Metriken, wie die bisher vergangene Zeit (unten rechts).

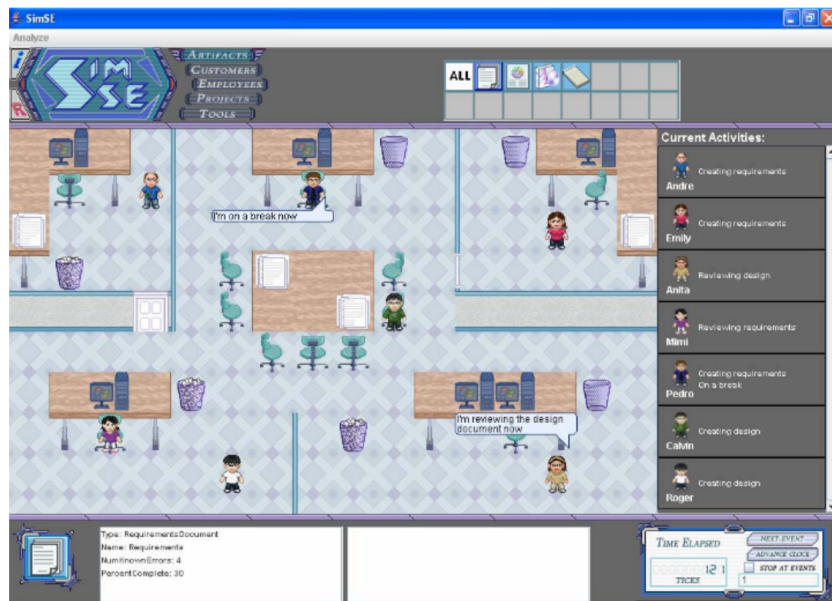


Abbildung 4: Screenshot aus SimSE (Navarro, 2006, S. 104)

### 2.3.2 Serious Toy Design

Der zweite Quadrant, das Serious Toy Design, wird von Deterding et al. beschrieben als die Erstellung von *paidic wholes*, also das Design oder Deployment von vollständigen Spielsachen für ernsthafte bzw. zielgerichtete Zwecke (Deterding et al., 2011a, S. 13).

**Beispiel:** Ein Beispiel stellt die von LEGO entwickelte *LEGO Mindstorms EV3* Technologie (LEGO Group, 2016) dar. Dabei handelt es sich um LEGO-Bausteine, was zunächst als Spielzeug aufgefasst werden kann, an die Motoren oder Sensoren angebaut werden, deren Verhalten durch eine oder mehrere Steuereinheiten (sog. *Bricks*) programmiert werden können. Für die Programmierung können entweder eine Hochsprache, wie z. B. Java genutzt werden oder die von LEGO bereitgestellte grafische Programmierumgebung. So kann Spielzeug dafür genutzt werden, abhängig von Alter bzw. Schwierigkeit, Programmierkonzepte zu lernen oder bereits vorhandene Kompetenzen in der objektorientierten Programmierung anzuwenden und weiterzuentwickeln. Die Abbildung 5 zeigt das Ergebnis eines studentischen Projekts<sup>6</sup> an der Hoch-

<sup>6</sup> An dieser Stelle sei dem Autor eine etwas unwissenschaftliche Anmerkung erlaubt. Für die im Jahresturnus stattfindenden studentischen Projekte, bei denen ein Einsatz von *LEGO Mindstorms* für Studierende erkennbar ist, ist eine Bewerberquote zu beobachten, die sich von anderen Projektangeboten deutlich abhebt (teilweise größer Faktor 8). Mit Sicherheit ist diese Tatsache nicht alleine durch die Kausalität zu erklären, dass dieses Lehr-Lernmittel zum Einsatz kommt. Dennoch – und das zeigt die Performanz der Studierenden während der Projektlaufzeit sowie die Evaluationen der Projekte – liegt hier zumindest ein einschlägiger Einfluss vor, der sich durch seine Regelmäßigkeit auffällig positiv darstellt.



schule Kempten aus dem Sommersemester 2016, welches vom Autor dieser Arbeit betreut wurde. Ziel des Projekts war es, mit LEGO Mindstorms EV3 einen Affen zu konstruieren (Abbildung 5 (a)) sowie für diese Tierart typische Fortbewegungsarten (vorwärts laufen (b), klettern (c), hangeln (d)) mit der Programmiersprache Java umzusetzen.

### 2.3.3 *Gameful Design*

Gameful Design<sup>7</sup> ist die Nutzung von *ludic qualities* im Design, um experimentelle und verhaltensbeeinflussende Qualitäten (im Sinne von Objekten, Werkzeugen und Kontexten; Nutzung dieser Qualitäten entspricht *gameful interaction* (Deterding et al., 2011a, S. 10)) zu erstellen, die charakteristisch für Games sind und *gamefulness*, also eine erlebte (Spiel-) Erfahrung beim Nutzer hervorrufen (Deterding et al., 2011a, S. 10).

Da es sich bei dem Konzept von Gamification mehr um einen zielgerichteten und zweckgebunden Stil des Spielens handelt, welcher kein gesamtes Lernspiel darstellt, sondern sich vornehmlich auf einzelne spielerische Elemente stützt und dabei vordergründig die Einstellung (z. B. durch stärkeres Involvieren bzw. Engagement) von Spielern (hier: Lernenden) versucht zu ändern, als am Spiel selbst zu lernen, ist es anhand dessen zu unterscheiden von dem Bereich der Serious Games (Bartel u. Hagel, 2016a; Danelli, 2015; Deterding et al., 2011a; Grünberger, 2014; Landers, 2014).

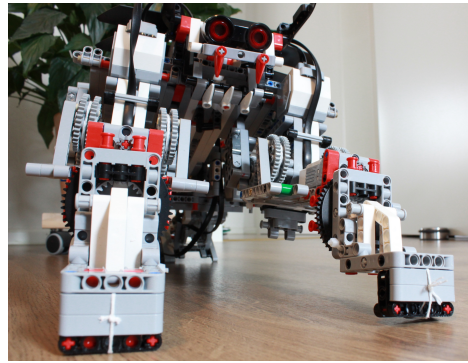
**Beispiel:** Ein Beispiel für Gameful Design nach Deterding et al. stellt die von Stöcklin et al. erstellte Plattform QuesTanja (Stöcklin et al., 2014) (Stöcklin et al., 2015) dar. Dabei handelt es sich um eine Gamification Online-Plattform (siehe Abbildung 6), die für den Unterricht<sup>8</sup> an Schulen entwickelt wurde. Unterrichtseinheiten können mit Game-Elementen (z. B. Quests mit Levels, Punkte und Leaderboards) strukturell angereichert werden und durch eine Storyline unterstützt, sollen Lernende dazu angeregt bzw. motiviert werden, sich selbständig mit Lerninhalten auseinanderzusetzen (Stöcklin et al., 2015). Bei dieser Unterteilung der Art wie spielifiziert wird, folgen die Macher von QuesTanja der Empfehlung von Kapp et al., welche zwischen *struktureller* und *inhaltlicher* Gamification im Lehr- Lernkontext unterscheiden (Kapp et al., 2014, S. 55f.). Während strukturelle Gamification darauf abzielt spiel-

<sup>7</sup> Deterding et al. sehen Gamification als eine Instanz von Gameful Design und somit Gameful Design in seiner Anwendung (Deterding et al., 2011a). Andere Forscher, wie z. B. Lee u. Doh widersprechen dieser These und sehen Gamification als eine Anwendung für extrinsische Motivation, während Gameful Design die intrinsische Motivation fokussiert (Lee u. Doh, 2012).

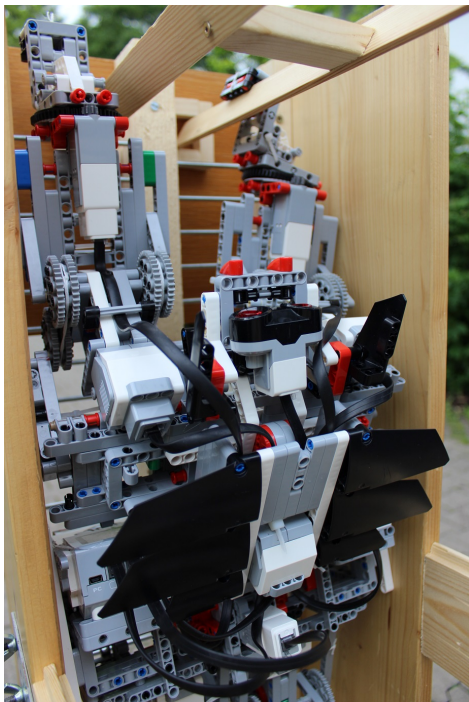
<sup>8</sup> In (Stöcklin et al., 2014) wird der Einsatz der Plattform im Matheunterricht beschrieben.



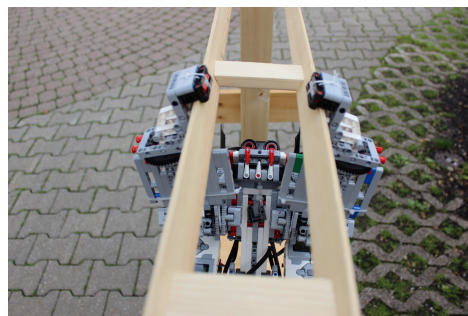
(a) Komplettansicht seitlich



(b) Frontalansicht



(c) Übergang von Klettern zu Hangeln



(d) Hangeln

Abbildung 5: LEGO Mindstorms EV3 Affe - Projekt an der Hochschule Kempten

typische Elemente (z. B. Badges, Leaderboards, Punkte) zu verwenden und dabei den (Lern-)Inhalt nicht verändert (Kapp et al., 2014, S. 224), wird bei inhaltlicher Gamification der (Lern-)Inhalt so adaptiert, dass beispielsweise

eine spieltypische Sprache verwendet wird, die durchaus kontextspezifisch sein soll und zudem „[...] game mechanics, and game thinking [...]“ (Kapp et al., 2014, S. 237) zur Anwendung kommen, „[...] to make it more game-like“ (Kapp et al., 2014, S. 237).



Abbildung 6: Screenshot der Lernplattform QuestTanja (Stöcklin et al., 2014, S. 272)

#### 2.3.4 Playful Design

Playful Design<sup>9</sup>, entspricht der Nutzung von *paidic qualities* im Design, um experimentelle und verhaltensbeeinflussende Qualitäten zu nutzen, die charakteristisch für Playing sind (Deterding et al., 2011a, S. 10).

**Beispiel:** Hier wird exemplarisch eine Tasse angeführt (siehe Abbildung 7), die als Aufdruck die Vorderseite eines *Nintendo Gameboys* besitzt. Dieses *Playful Design* (Ferrara, 2012) von Alltagsgegenständen besitzt idealerweise eine hohe *affordance*<sup>10</sup> (Norman, 1995), also eine hohe Selbsterklärungsfähigkeit in Bezug auf seine spielerischen Eigenschaften und damit konnotierten Verbindungen von dem jeweiligen Objekt, zu konkreten Handlungen durch den Menschen (Gibson, 1979).

<sup>9</sup> Auch wenn Deterding et al. *Playful Design* begrifflich von *Gameful Design* distanzieren, schließen sie nicht aus, dass eine spielifizierte Anwendung nicht auch spielerische Verhalten im Sinne von *play* hervorrufen kann (Deterding et al., 2011a). Dies würde damit einem Abdriften in ein *paidiasches* Spiel entsprechen, in dem der Nutzer ausschließlich zum Unterhaltungszweck spielt.

<sup>10</sup> Obwohl der Begriff *affordance* vermehrt im Usability bzw. User-Interface-Design anzutreffen ist, lässt er sich auch auf Bereiche übertragen, die keinen direkten Bezug zu seinen regulären Anwendungsfeldern besitzen. Dieser Adaption wird dadurch Rechnung getragen, dass der Wahrnehmungspsychologe Gibson die ursprüngliche und erstmalige Definition des Begriffs unabhängig von den heutigen Anwendungsdomänen erstellte (Gibson, 1979).



Abbildung 7: Playful Design bei Alltagsgegenständen (Grotton, 2012)

### 2.3.5 Synonymische Verwendungen

Deterding et al. merken an, dass sich neben dem Begriff *Gamification* auch weitere Begriffe, wie z. B. *Productivity Games* (McDonald et al., 2007), *Funware* (Takahashi, 2008) oder *Behavioral Games* (Dignan, 2011) synonymisch<sup>11</sup> verwendet werden, sich jedoch *Gamification* als Begriff durchgesetzt hat (Deterding et al., 2011a). Dies zeigt deutlich die Begriffsentwicklung<sup>12</sup> (siehe Abbildung 8) und die damit verbundene starke Verbreitung<sup>13</sup> des Konzepts zum Jahreswechsel von 2010 auf 2011, gemessen an der Datenbasis von Google-Suchanfragen. Die Abbildung 8 zeigt einen normalisierten Vergleich der Anzahl an Suchanfragen eines Begriffs zu der Anzahl an gesamten Suchanfragen über den gesamten Darstellungszeitraum. Ein Wert von 0 sagt also aus, dass marginal wenige bzw. keine Suchanfragen zu einem Begriff über den Zeitraum getätigt wurden. Dementsprechend würde ein Wert von 100 bedeuten, dass der Begriff am häufigsten, im Vergleich zu anderen Begriffen, innerhalb des definierten Zeitraums, gesucht wurde. Sicherlich ist die genutzte Datenbasis nicht alleinig repräsentativ für die tatsächliche Entwicklung eines Forschungsfelds, vermittelt aber in diesem Fall einen Eindruck, der so in der wissenschaftlichen Literatur deutliche Bestätigung (siehe z. B. Boulet 2012; Deterding et al. 2011a; Werbach 2014) findet.

Die tendenziell abfallende Suchhäufigkeit der Serious Games und der gleichzeitige rasante Anstieg von Gamification sollte nicht losgelöst voneinander interpretiert werden. Durch die bereits beschriebene Begriffsheterogenität ist

<sup>11</sup> Weitere jedoch eher punktuell in der Literatur eingesetzte Synonyme sind *Alternate Reality Games* (McGonigal, 2011; McGonigal, 2012), *Persuasive Games* (Bogost, 2011), *Human-Focused Design* (Chou, 2015) oder eher kritischere Bezeichnungen, wie *Pointsification* (Robertson, 2010) oder *Exploitationware* (Bogost, 2011).

<sup>12</sup> Erstellt mit Google Trends: <https://www.google.com/trends>, abgerufen am: 10.08.2015

<sup>13</sup> Nach Knaving et al. wurde der Begriff *Gamification* erstmalig 2008 verwendet, obwohl das Konzept an sich mindestens seit 1980 existiert (Knaving et al., 2013, S. 132). Jedoch wurde dem Begriff nicht die nötige Aufmerksamkeit entgegen gebracht, bis Fachkonferenzen und die Industrie die Nutzung des Begriffs maßgeblich vorantrieben (Groh, 2012).

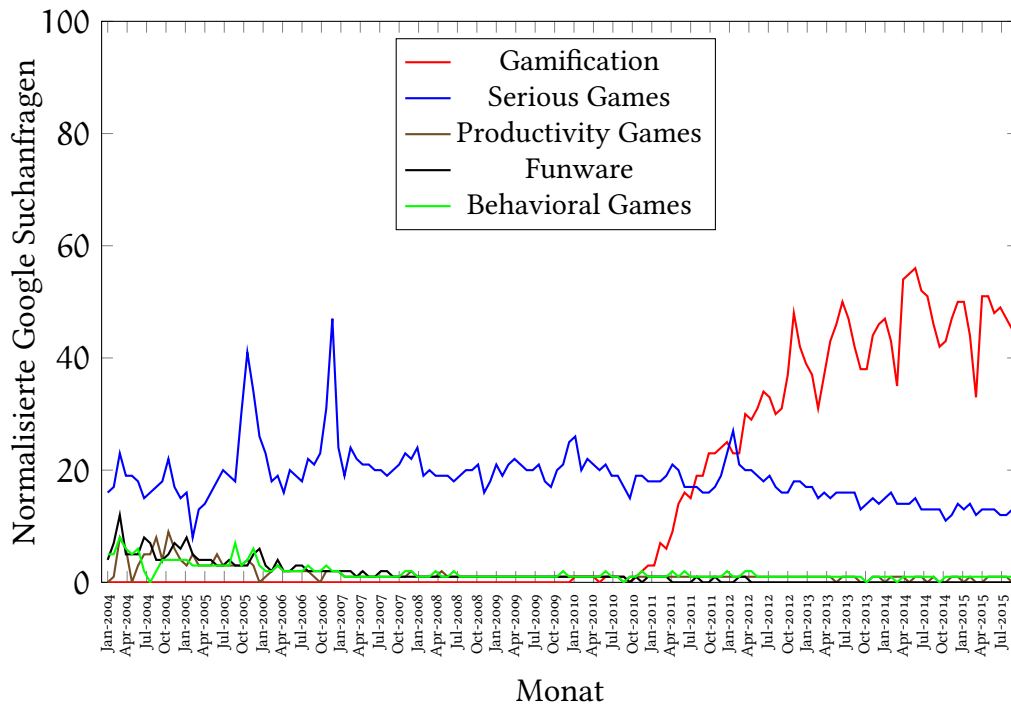


Abbildung 8: Google-Suchanfragen zu Gamification und synonymisch verwendeten Begriffen im zeitlichen Verlauf (Jan. 2004 - Jul. 2015)

es durchaus denkbar, dass der damals neu aufkommende Begriff Gamification als Trendbegriff verwendet wurde, um auch solche Anwendungen zu betiteln oder zu charakterisieren, die vor dem Aufkommen des Begriffs, den Serious Games zuzuordnen waren (Göbel et al., 2014; Seaborn u. Fels, 2015). Kapp unterstellt sogar, dass es sich bei Serious Games, um ein Subset von Gamification handelt, in dem speziellen Sinne, dass Gamification eine Art und Weise darstellt, sich in Serious Games zu manifestieren (Kapp, 2012b). Dies mag der gemeinsamen Basis, dem Game Design und dessen Taxonomie für Game Design-Elemente, geschuldet sein (Deterding et al., 2011a; Landers, 2014; Werbach, 2014).

## 2.4 GAMIFICATION DESIGN-ELEMENTE

Im Folgenden werden Gamification Design-Elemente definiert und begrifflich eingeordnet. Zudem erfolgt eine auf einschlägiger Literatur basierte Vorstellung etablierter Schemata zur Klassifikation derartiger Elemente.



### 2.4.1 Begriffliche Annäherung

Zunächst soll klar umrissen werden, was Gamification bezogen auf seine Elemente ausmacht. Unter Gamification Design-Elementen verstehen Seaborn u. Fels „[...] patterns, objects, principles, models, and methods directly inspired by games“ (Seaborn u. Fels, 2015, S. 8). Dies widerspricht dem Begriffsverständnis von Bjork u. Holopainen, die derartige Design Elemente verstehen als „[...] the physical and logical components that contain the game state and are manipulated by players to achieve their goals“ (Bjork u. Holopainen, 2005, S. 26). Es zeigt zudem, die ebenfalls starke Heterogenität im Begriffsverständnis eines *Game*, welche respektiv bei Gamification vorzufinden war (siehe hierzu z. B. Bjork u. Holopainen (2005, S. 7)). Der Widerspruch zwischen den Forschern ist begründbar durch den unterschiedlichen Blickwinkel, der bei einer Betrachtung von Game Design-Elementen existiert: Seaborn u. Fels, welche die Perspektive aus einer Gamification Sicht einnehmen, stellen diese einer aus der klassischen Game Design stammenden Sicht durch Bjork u. Holopainen gegenüber. In diesem Definitionskonflikt manifestiert sich gleichzeitig die Verbundenheit beider Domänen, aber ebenso die Notwendigkeit einer Unterscheidung.

Deterding et al. hingegen, geben an, dass es sich bei Game Design-Elementen um eine Sammlung bestehend aus „[...] building blocks or features shared by games (rather than a set of necessary conditions for a game)“ handelt (Deterding et al., 2011a, S. 12). Weiterhin erläutern sie, dass diese Art der Definition auf zwei Arten unterschiedlich interpretiert werden kann. Eine strenge Interpretation würde bedeuten, dass die Definition nur auf solche Game-Elemente abzielt, welche einerseits einzigartig oder andererseits spezifisch für Games sind (Deterding et al., 2011a, S. 12). Die Ergebnismenge einer solchen Betrachtung wäre sehr klein und homogen. Eine sehr liberale Betrachtungsweise hätte zur Folge, dass jedes Element welches in jedem Game identifiziert werden kann, durch diese Definition angesprochen wird (Deterding et al., 2011a, S. 12). Letztere Interpretation würde zu einer enorm großen Ergebnismenge führen, wobei man nicht unbedingt davon ausgehen kann in diesem Fall von einer Menge zu sprechen, da diese die Charakteristik der Abgeschlossenheit besitzt (Huotari u. Hamari, 2012, S. 19). Schließlich schlagen sie vor, Gamification bezogen auf seine Elemente in Form eines Mittelwegs aus beiden Interpretationsmöglichkeiten zu umreißen, in dem sie formulieren:

[...] restricting Gamification to the description of elements that are characteristic to games - elements that are found in most (but

not necessarily all) games, readily associated with games, and found to play a significant role in gameplay. (Deterding et al., 2011a, S. 12)

Als eine Einschränkung geben sie zu bedenken, dass es vonnöten ist ein genaueres Verständnis davon zu bekommen, was charakteristisch für Games bedeutet. Für die vorliegende Arbeit soll zum Zwecke der begrifflichen Schärfung eine Einschränkung getroffen werden, die es erlaubt jene Game-Elemente, die nach der oben vorgestellten Definition von Deterding et al. charakteristisch für Games sind, von weiteren Game-Elementen zu unterscheiden, die im Gamification-Ansatz vornehmlich vorzufinden sind. Die für Gamification spezifischen Game-Elemente, sollen gemäß ihrer Spezifität fortan als Gamification Design Elemente (GDE) bezeichnet werden (Bartel et al., 2015). Um diese Spezifität genauer zu charakterisieren, existieren in der Literatur unterschiedliche Klassifikationsschemata, die es erlauben, Game-Elemente allgemein zu klassifizieren. Die Grundlage für diese Einschränkung von Game-Elementen auf GDE bildet das Klassifikationsschema (*Gamification Toolkit* (Werbach u. Hunter, 2015)) von Werbach u. Hunter, welches neben weiteren im Folgenden näher ausgeführt werden soll.

#### 2.4.2 Klassifikationsschemata

Für das Design von Gamification existiert kein weitläufig anerkanntes und wohldefiniertes Set an Game Design-Elementen (Dicheva et al., 2015, S. 3; Huotari u. Hamari, 2012, S. 19; Hamari et al., 2014, S. 3026). Um diese These zu untermauern, stellen Dicheva et al. beispielhaft dar, wie das Game Element *Badge* in bekannten Klassifikationsschemata eingeordnet werden würde bzw. wie eine Einordnung bereits erfolgt ist (Dicheva et al., 2015, S. 3). Demnach wäre ein *Badge* nach Deterding et al. ein *Game Interface Pattern* (Deterding et al., 2011a), ein *Game Mechanic* in der Unterteilung von Zichermann u. Cunningham (Zichermann u. Cunningham, 2011), ein *Game Dynamic* nach Ansicht von Iosup u. Epema (Iosup u. Epema, 2014), eine *Game Technique* in der Taxonomie von Chou (Chou, 2015), eine *Motivational Affordance* in den Augen von Hamari et al. (Hamari et al., 2014) und schließlich ein *Game Component* nach Interpretation von Werbach u. Hunter (Werbach u. Hunter, 2012). Dicheva et al. fügen weiterhin hinzu, dass sich die genannten Autoren nicht nur in der Art der Eingruppierung und Bezeichnung unterscheiden, sondern dass die Gruppierungen zudem auf unterschiedlichen Abstraktionsebenen definiert werden, was eine einheitliche Betrachtungsweise unmöglich macht (Dicheva et al., 2015, S. 3). Trotz der aufgezeigten Heterogenität, die sich auch durch diesen Bestandteil der allgemeinen Game Theorie durchzieht, sollen im

Folgenden, wichtige Klassifikationsschemata und deren Charakteristika kurz vorgestellt werden.

#### 2.4.2.1 *Deterdings Game Design-Elemente-Taxonomie*

Deterding et al. sehen in einer weiteren Unterteilung von Game-Elementen, in Bezug auf das Design, einen direkten Bezug für Gamification. Sie beschreiben eine Taxonomie, welche fünf verschiedene Abstraktionslevel (AL) von Game Design-Elementen enthält (siehe Tabelle 1).

AL	Name	Beschreibung	Beispiel
1	Game Interface Design Patterns	Gängige und erfolgreich eingesetzte Interaktionsdesignkomponenten und Designlösungen für ein bekanntes Problem in einem bestimmten Kontext (einschließlich der prototypischen Umsetzung).	Badge, Leaderboard, Level (siehe z. B. Bartel u. Hagedorn (2014); Stöcklin et al. (2014))
2	Game Design Patterns und Mechanics	Gewöhnlich wiederauftretende Teile des Game Designs, welche sich auf das Gameplay an sich beziehen	Zeitvorgaben, eingeschränkte Ressourcen, Turns (siehe z. B. Zichermann u. Cunningham (2011); Bjork u. Holopainen (2005))
3	Game Design Prinzipien und Heuristiken	Evaluative Richtlinien um eine Designfragestellung anzugehen oder eine bestehende Designlösung zu analysieren	Fortwährendes Spielen, klare Ziele, Vielfalt von Game Stilen (siehe z. B. Danelli (2015); Schell (2015))
weiter auf nächster Seite			



AL	Name	Beschreibung	Beispiel
4	Game Models	Konzepte der Game-Komponenten oder der Game-Wahrnehmung bzw. des „Erlebens“ von Games	Mechanic-Dynamic-Aesthetics (Zichermann u. Cunningham, 2011), Fantasy-Challenge-Curiosity (FCC) (Malone u. Lepper, 1987), Game Design Atoms (Deterding, 2013b), Core Elements of Game Experience (CEGE) (Calvillo-Gómez et al., 2010)
5	Game Design Methods	Für Game Design spezifische Handlungsempfehlungen und Prozesse	Playtesting, spielerzentriertes Design, Wertgebundenes Game Design (siehe z. B. Salen u. Zimmerman (2004); Schell (2015); Rehfeld (2014))

Tabelle 1: Taxonomie von Game- Design-Elementen (Deterding, 2011a, S. 12)

Durch diese Form des fortwährend abstrakteren Aufbaus der Game Design-Elemente auf den unterschiedlichen Abstraktionsniveaus wird deutlich, dass es beispielsweise nicht möglich ist, „fortwährendes Spielen“ (AL<sub>3</sub>) explizit sowohl durch den Designer als auch durch den Nutzer innerhalb eines Games anzusprechen oder gar als Komponente bei dem Design einzusetzen. Deshalb ist es nicht möglich, diese aggregierten Konzepte direkt in das Design von Gamification als Komponente mit einzubauen. Sie äußern sich in ihrem „Erleben“ nur indirekt über den systematischen Aufbau aus den Bestandteilen der weniger abstrakteren Abstraktionslevel (Werbach u. Hunter, 2012, S. 78 ).

### 2.4.2.2 Schells Tetrade der Game-Elemente

Bei der Tetrade der Game-Elemente (engl. *Elemental Tetrad* (Schell, 2015)) handelt es sich um ein weiteres Klassifizierungsschema von Game-Elementen zur Erstellung von Games, welches Game-Elemente unterteilt in: *Mechanics*, *Aesthetics*, *Story* und *Technology* (siehe Abbildung 9).

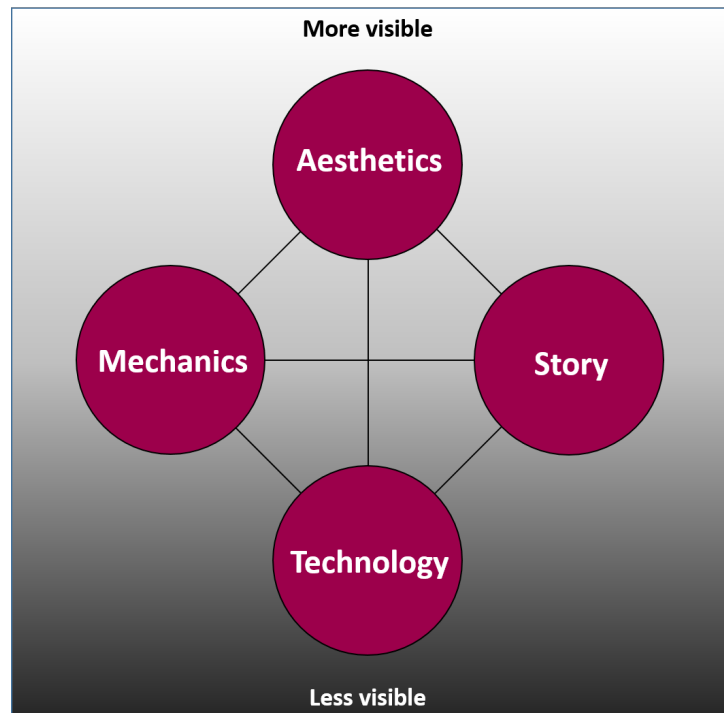


Abbildung 9: Schells Tetrade der Game-Elemente (Schell, 2015, S. 51)

**MECHANICS** Schell versteht unter *Mechanics* alle Prozeduren und Regeln im Spiel, also die Ziele des Games, unter welchen Bedingungen Spieler diese erreichen können und was passiert, wenn sie es versuchen (Schell, 2015, S. 51):

„Gameplaying is decision making. Decisions are made based on information. Deciding the different attributes, their states, and what changes them is core to the mechanics of your game“ (Schell, 2015, S. 167).

Beispiele sind *Space* (Schell, 2015, S. 158ff.), *Time* (Schell, 2015, S. 163ff.), *Objects*, *Attributes*, *States* (Schell, 2015, S. 165ff.) und *Actions* (Schell, 2015, S. 170ff.).

Schell merkt außerdem an, dass *Mechanics* durch *Technology* unterstützt werden. *Mechanics* sind entlang einer *Story* eingebettet und werden dabei

durch *Aesthetics* entsprechend für Spieler zur Geltung gebracht (Schell, 2015, S. 52).

**STORY** Unter dem Element *Story* versteht Schell die Reihenfolge an Ereignissen, die innerhalb eines Games auftreten. Er unterscheidet zwischen einer linearen (Schell, 2015, S. 298) und verzweigten Story (Schell, 2015, S. 302) und nennt für beide Typen die Notwendigkeit diese durch *Mechanics* zu unterstützen, durch *Aesthetics* kognitiv aufbereiten und durch *Technology* entsprechend der Art der Story anzureichern (Schell, 2015, S. 52).

**AESTHETICS** Das Aussehen, der Klang, der Geruch, der Geschmack, und das Gefühl eines Games bilden einen zentralen Aspekt für Schell, da sie die Sinne eines Menschen nutzen, um die Verbindung zwischen einem Spieler und dem Game herstellen (evtl. über Bedienelemente und Interfaces) (Schell, 2015, S. 52). Auch bei diesen Elementen bedarf es einer Unterstützung der anderen drei Game-Elemente, die das Erleben eines Games für einen Spieler immersiver im Sinne von „[...] solid, real, and magnificent [...]“ (Schell, 2015, S. 385) machen.

**TECHNOLOGY** Für diesen Elementtyp schließt Schell explizit eine Nutzung von nicht hoch-technologischen Werkzeugen nicht aus (Schell, 2015, S. 52). So ist es genauso möglich neben „technology-rich platforms“ vergleichsweise traditionelle Medien wie Papier oder Stifte zu verwenden, so lange diese ein Spielen ermöglichen (Schell, 2015, S. 52). Dieser Elementtyp ist essentiell, denn dadurch „[...] the aesthetics take place, [...] the mechanics will occur, and through which the story will be told“ (Schell, 2015, S. 52). Zudem ist es auch das „[...] most dynamic, most volatile, and most unpredictable“ (Schell, 2015, S. 450) Element, welches es erschwert, sich auf das Game Design zu konzentrieren, sobald dieses Element die initiale Aufmerksamkeit im Zuge einer Spielerstellung erfahren hat (Schell, 2015, S. 450).

Die vier in der Tetrade befindlichen Elemente sind in einem stabilen Gleichgewicht angeordnet, so dass kein Element wichtiger ist als ein anderes und stehen in gegenseitiger Wechselwirkung (Schell, 2015, S. 52). Jedes der Elemente muss im Zuge eines Game Designs betrachtet werden. Jedoch sind manche Elemente für Spieler sichtbarer als andere (Schell, 2015, S. 52).

### 2.4.2.3 Hunickes Mechanics-Dynamics-Aesthetics-Framework

Ein weiteres Modell zur formalen Kategorisierung von Game Design-Elementen ist das Mechanics-Dynamics-Aesthetics-Framework (siehe Abbildung 10) (Hunicke et al., 2004). Hunicke et al. rücken explizit Games in den Fokus und nicht Gamification. Dennoch ist eine Adaption auf den Gamification-Bereich, wie beispielsweise in Zichermann u. Cunningham (2011) erfolgt, möglich. Jedoch merken Hunicke et al. an, dass das Framework in erster Linie als diagnostisches Framework zur Analyse von Games gedacht ist (Hunicke et al., 2004, S. 1), was durch den Begriff der „postmortem analysis“ (Zichermann u. Cunningham, 2011, S. 35) von Zichermann u. Cunningham aus der Sicht von Gamification bestätigt wird.

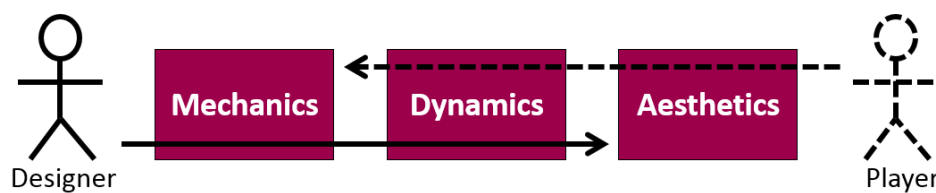


Abbildung 10: Hunickes Mechanics-Dynamics-Aesthetics-Framework (vgl. Hunicke et al., 2004, S. 2)

Hunicke et al. unterteilen das Framework in *Mechanics*, *Dynamics* und *Aesthetics* (Hunicke et al., 2004, S. 2 eigene Übersetzung):

- *Mechanics* beschreiben die speziellen Komponenten eines Games, auf dem Level der Repräsentation von Daten und Algorithmen. Rehfeld postuliert hierzu, dass es sich bei *Mechanics* in erster Linie um formale Regeln handelt, „[...] in denen der Game Designer die Rollen der Gamebits<sup>14</sup> und deren Relationen festlegt und regelt“ (Rehfeld, 2014, S. 72).
- *Dynamics* beschreiben das Laufzeitverhalten von *Mechanics* als eine Reaktion auf Aktionen durch den Spieler und untereinander von Spielern, während gespielt wird. Aus der Kombination aus *Mechanics* und *Dynamics*, so Rehfeld, entsteht ein System, mit dem der Spieler interagiert<sup>15</sup> und das er wahrnimmt (Rehfeld, 2014, S. 72).
- *Aesthetics* beschreiben die erhofften emotionalen Reaktionen, die in einem Spieler während des Spielens hervorgerufen werden sollen. Schell

<sup>14</sup> Rehfeld versteht unter dem Begriff Gamebits Elemente des Spiels, die nicht Regeln und das visuelle Interface umfassen (siehe hierzu Rehfeld (2014, S. 13, 42)).

<sup>15</sup> Norman stellt hierzu ein Schema bereit (Norman, 2002), wie Benutzer mit der Umwelt interagieren, welches auf Games übertragen werden kann und in Rehfeld (2014, S. 82) ausführlich diskutiert wird.

formuliert zu diesem kognitiven Aspekt ergänzend: „This is how your game looks, sounds, smells, tastes, and feels“ (Schell, 2015, S. 52). Rehfeld folgt der Interpretation von Hunicke et al. und stellt den Spaß<sup>16</sup> als das Ziel der *Aesthetics* dar, was durch Komponenten wie das Eintauchen in Traumwelten, eine interessante Erzählung oder das Beweisen eines Spielers in Wettkämpfen begünstigt werden soll (Hunicke et al., 2004, S. 2; Rehfeld, 2014, S. 73f.).

Zentral an diesem Ansatz ist nach Hunicke et al., dass jede Komponente in dem Framework als eine Sicht (engl.: lense) auf ein Game gesehen werden kann (Hunicke et al., 2004, S. 2). Dies bedeutet, dass je nach Perspektive die Komponenten unterschiedlich wahrgenommen werden. So sieht ein Game Designer zunächst die *Mechanics* als für ihn relevant an, die eine starke Bindung zu den *Dynamics* aufweisen, da letztere das Laufzeitverhalten darstellen und schließlich zu ästhetischen und wahrnehmbaren Erfahrungen führen (Hunicke et al., 2004, S. 2). Für den Spieler stellen die *Aesthetics* die in erster Linie wahrnehmbare Elemente von Games dar, welche Emotionen bei Spielern hervorrufen sollen (Zichermann u. Cunningham, 2011, S. 36). Welche Emotionen jedoch erzeugt werden, differiert von Spieler zu Spieler stark: „Die empfundene Wahrheit des Game Designers muss nicht die Empfindung (Wahrheit) der Spieler sein“ (Rehfeld, 2014, S. 73). Deshalb sehen Hunicke et al. eine Notwendigkeit, dass Game Designer und Spieler, während der Entwicklung eines Games, in eine starke Wechselwirkung zueinander treten, so dass eine ständige Rückkopplung (z. B. durch die Technik des Playtesting (Schell, 2015, S. 433ff.)) von Spieler zu Game Designer erfolgen kann und damit die Möglichkeit von Anpassungen während dem Game Design gegeben ist.

#### 2.4.2.4 Werbach und Hunters Gamification Toolkit

Auch Werbach und Hunter nehmen sich der Definition von Game-Elementen an. Für sie manifestiert sich ein Game durch ein integriertes Erleben, welches auf die einzelnen Bestandteile, die Game-Elemente zurückzuführen ist (Werbach u. Hunter, 2012, S. 26). Sie vergleichen die Game-Elemente mit einem zweckgerichteten Baukasten, welcher für die Erstellung von Games benutzt werden kann, geben aber wie Deterding et al. zu bedenken, dass die Elemente für Gamification nicht alle Elemente eines Games umfassen können (Werbach u. Hunter, 2012, S. 27). Daraus schließen sie eine höhere Flexibilität bei dem Design von Gamification im Vergleich zu Games in Bezug auf den Einsatz der

<sup>16</sup> Für eine detaillierte Analyse zu den Gründen und Formen des menschlichen Spielspaßes, sei an dieser Stelle auf Koster (2013) verwiesen.

jeweiligen Elemente, was durch die Tatsache begründet wird, dass „[...] game elements can be embedded into activities that are not themselves games“ (Werbach u. Hunter, 2012, S. 27). Nach Werbach u. Hunter gibt es drei Kategorien von Game-Elementen mit unterschiedlichem Abstraktionsgrad, welche für Gamification von Relevanz sind (siehe Abbildung 11): *Dynamics*, *Mechanics* und *Components* (Werbach u. Hunter, 2012, S. 78).

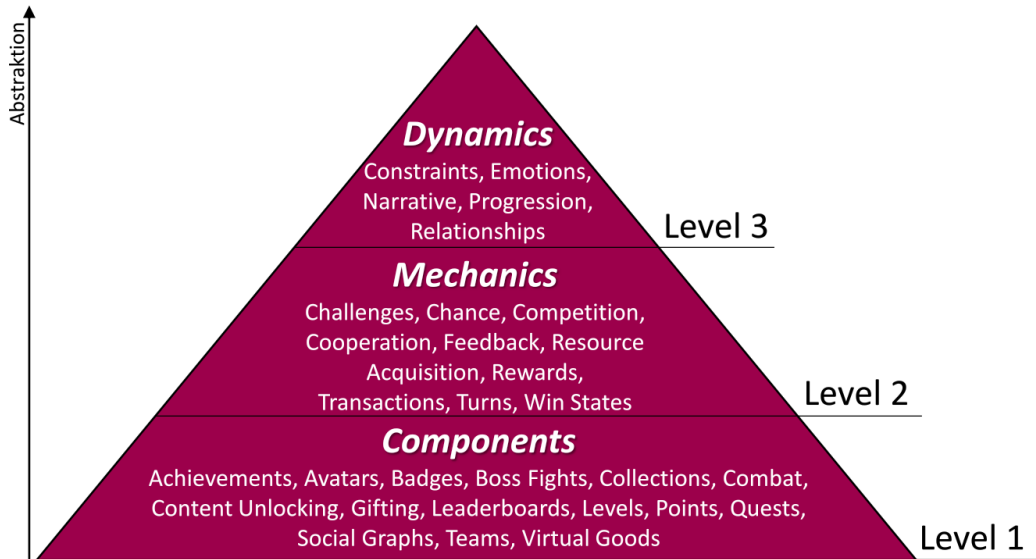


Abbildung 11: Gamification Toolkit nach Werbach u. Hunter (2015, S. 22ff.)

Dabei besteht jede *Dynamic* aus mindestens einer *Mechanic*, wohingegen jede *Component* an eine oder mehrere abstraktere Elemente gebunden ist (Werbach u. Hunter, 2012, S. 78). Anders gesprochen realisiert eine oder mehrere *Mechanic(s)* eine *Dynamic*, wohingegen die visuelle und damit wahrnehmbare Repräsentation einer *Mechanic* eine oder mehrere *Component(s)* darstellt bzw. darstellen (Bartel u. Hagel, 2016b, S. 75). Auch kann die Ebene der *Mechanics* in gewisser Weise übersprungen werden, indem eine *Dynamic* durch eine oder mehrere *Component(s)* realisiert und für den Benutzer sichtbar gemacht wird bzw. werden (Bartel u. Hagel, 2016b, S. 75). Abbildung 12 verdeutlicht grafisch diese Beziehung der Elemente einzelner Abstraktionsebenen zueinander.

**DYNAMICS** Die Kategorie mit dem höchsten Abstraktionsgrad sind die *Dynamics*. Nach Werbach u. Hunter handelt es sich dabei um

„[...] big-picture aspects of gamified systems that you have to consider and manage but which you can never directly enter into the game“ (Werbach u. Hunter, 2012, S. 78).

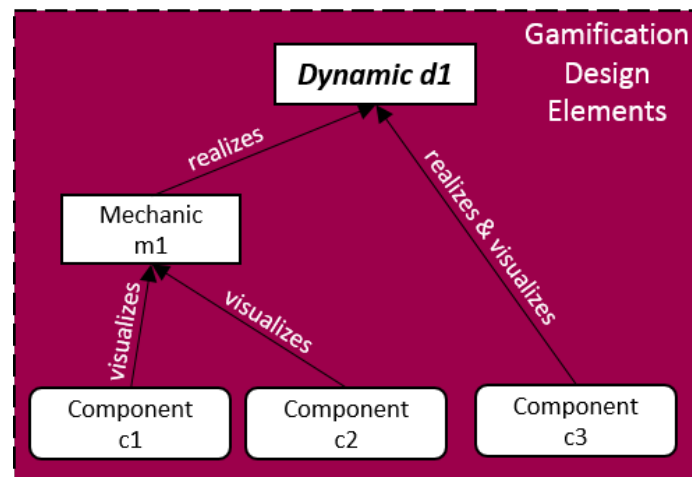


Abbildung 12: Zusammenhang von Dynamics, Mechanics und Components (Auszug aus Bartel u. Hagel (2016b, S. 75) bzw. Werbach u. Hunter (2015, S. 22))

Zichermann u. Cunningham fügen hinzu, dass Dynamics als die Art und Weise beschrieben werden, wie Spieler mit „Game Experiences“ interagieren (Zichermann u. Cunningham, 2011, S. 77). Dabei reduzieren sie die *Dynamics* auf eine Interaktion des Spielers mit den *Mechanics* und folgen damit sowohl der Argumentation von Hunicke et al. und ihrem MDA-Ansatz, als auch der von Werbach u. Hunter, die das Erzeugen bzw. Adressieren von Emotionen als eine Subkomponente der *Dynamics* sehen (Rehfeld, 2014, S. 72; Werbach u. Hunter, 2012, S. 78; Zichermann u. Cunningham, 2011, S. 36). Zu den *Dynamics* zählen Werbach u. Hunter beispielsweise die Gamification Elemente *Emotionen*, *Fortschritt* und *Beziehungen* (Werbach u. Hunter, 2015, S. 25ff.).

**MECHANICS** Jede *Mechanic* ist eine Art, mindestens ein *Dynamic* zu erreichen, welches einen direkten Bezug zur *Mechanic* hat (Werbach u. Hunter, 2012, S. 80). Dabei bezieht sich jede *Mechanic* auf eine Regel oder ein Set an Regeln, welches den Handlungsspielraum eines Spielers einschränken oder ausweiten, indem sie eine Beziehung der Art „Grund für ein Handeln - erzielter Effekt durch das Handeln“ beschreiben (Kapp et al., 2014, S. 96). Ähnlich argumentiert Schell, indem er die *Mechanics* als eine Zusammenfassung von Prozeduren und Regeln eines Spiels versteht (Schell, 2015, S. 51). Genauer gesagt beschreiben die *Mechanics* das Ziel eines Spiels, was Spieler dafür tun müssen, um dieses Ziel zu erreichen bzw. nicht zu erreichen, und was passiert, wenn sie es versuchen (Schell, 2015, S. 51).

Bezogen auf das Lernen, so folgern Niman und Dichev et al., sind die *Mechanics* kein Ersatz für gutes Game Design. Vielmehr sind diese der Akteur,

welcher gutes Game Design in eine lebhafte und bedeutende Lernerfahrung umwandelt (Dichev et al., 2014, S. 90f.; Niman, 2014, S. 122).

**COMPONENTS** Bei den *Components* handelt es sich um eine atomare Einheit, welche von Elementen aus *Dynamics* oder *Mechanics* jeweils angenommen werden kann (Werbach u. Hunter, 2015, S. 43). Jedoch können die einzelnen Elemente innerhalb dieser Abstraktionsebene miteinander in Beziehung stehen (Bartel u. Hagel, 2016b, S. 75). Beispielsweise können *Leaderboards* gewisse *Points* als Metrik nutzen, um die einzelnen Spieler in einer entsprechenden Reihenfolge anzuordnen (Werbach u. Hunter, 2015, S. 47f.).

Die vorgestellten Schemata tragen dazu bei, einen für sich genommenen einheitlichen Blick auf Game Design-Elemente und damit wichtige, strukturelle Bausteine von Gamification zu erhalten. Speziell das zuletzt fokussierte Gamification Design Toolkit von Werbach u. Hunter (2015) soll in dieser Arbeit zur Anwendung kommen und stellt damit eine Grundlage für weitere Forschungsarbeit dar.

## 2.5 KOMPETENZENTWICKLUNG DURCH GAMIFICATION DESIGN-ELEMENTE

Im Folgenden wird das dieser Arbeit zu Grunde gelegte Verständnis von kompetenzorientierter Lehre erläutert und eine Theorie entwickelt sowie validiert, mit der sich die Domäne der kompetenzorientierten Lehre mit dem Gamification-Ansatz, im Speziellen den Dynamics, verbinden lässt.

### 2.5.1 Merkmale kompetenzorientierter Lehre

Anhand des bereits diskutierten Gamification Design Frameworks von Werbach u. Hunter (2015) zur Unterteilung von Game Design-Elementen konnte bezogen auf dessen Dynamics gezeigt werden, dass diese sich für eine strukturierte Zielformulierung für Kompetenzentwicklung eignen (Bartel et al., 2015). Dabei wurde zunächst einschlägige Literatur konsultiert und der Frage nachgegangen, inwiefern Dynamics zur Kompetenzentwicklung in Lehr-Lernkontexten Beachtung finden. Es wurde festgestellt, dass „[...] the relation of dynamics to implicitly or partly related competency-based education is not the centre of recent scientific debates“ (Bartel et al., 2015, S. 459). Bezogen auf den Kompetenzbegriff wurden Prinzipien erarbeitet, welchen diesen operationali-



sieren und dabei an das Begriffsverständnis von Weinert (2001) anlehnt. Nach Weinert handelt es sich bei Kompetenzen um

„[...] die bei Individuen verfügbaren oder durch sie erlernbaren kognitiven Fähigkeiten und Fertigkeiten, um bestimmte Probleme zu lösen, sowie die damit verbundenen motivationalen, volitionalen und sozialen Bereitschaften und Fähigkeiten, um die Problemlösungen in variablen Situationen erfolgreich und verantwortungsvoll nutzen zu können“ (Weinert, 2001, S. 27f.).

Damit wird klar, dass Kompetenzen nicht alleine kognitive Substanz besitzen. Vielmehr spielen ebenso psychosoziale Komponenten eine maßgebliche Rolle. Es erscheint vor allem beachtlich, dass Kompetenzen eine starke Kontextabhängigkeit besitzen und damit nur domänenspezifisch beurteilbar sind. Daraus folgt, dass eine objektive Messbarkeit auch für die psychosoziale Komponente(n) von Kompetenz gegeben sein muss. An dieser Stelle distanziert sich der Autor dieser Arbeit davon, psychosoziale Kompetenzanteile zu messen. Vielmehr soll der im Folgenden beschriebene Ansatz auf den vergleichsweise gut messbaren Teil von Kompetenz abzielen, der Fachkompetenz. Nebenbei, und das ist bewusst so gewählt, sollen psychosoziale Anteile fassbarer gemacht werden, indem spielerische Design Elemente zur Indikation eingesetzt werden. Zunächst soll jedoch umschrieben werden, welches Verständnis von kompetenzorientiertem Lernen der Arbeit von Bartel et al. (2015) zu Grunde liegt.

Die in darin beschriebene Sicht basiert auf dem folgenden Verständnis kompetenzorientierten Lernens und dessen unterschiedlichen Dimensionen<sup>17</sup> (Weinert, 1999, S. 4ff.; Bönsch et al., 2010, S. 29ff.; Erpenbeck, 2014, S. 21; Mandl u. Hense, 2014, S. 28):

- Kompetenzorientiertes Lernen ist ein aktiver, interaktiver und gleichzeitig konstruktiver Prozess, der selbstgesteuert-individuell verläuft und der Bedeutungserzeugung dient.
- Inhalt des Prozesses ist nicht allein der Erwerb von Wissen, sondern vielmehr die Verinnerlichung von fachspezifisch-methodischer, personaler und sozialer Kompetenzen, die in einschlägigen Situationen zum Vorschein treten (Performanz).
- Der Lernprozess ist ein emotionaler Prozess, untrennbar verbunden mit (intrinsischer) Motivation als zwingende Voraussetzung zur erfolgreichen Kompetenzentwicklung.

<sup>17</sup> Zitiert und übersetzt nach Bartel et al. (2015, S. 459).

- Kompetenzorientiertes Lernen bedeutet Lernen in Situationen, die Fehler zulassen ohne dabei das Selbstwirksamkeitsempfinden von Lernenden zu verringern und die dabei von Lehrenden begleitet werden.

### 2.5.2 *Dynamics als Brücke zur Kompetenzorientierung*

Bei der Gestaltung von kompetenzorientierten Lehr-Lernarrangements sehen Bartel et al. Anknüpfungsmöglichkeiten, wie sich Game Design-Elemente, vor allem Dynamics, anhand der Taxonomie von Werbach u. Hunter (2015) integrieren lassen (Bartel et al., 2015). Auch wenn Elemente dieser Taxonomiestufe sich nur über darunterliegende Stufen in spielifizierte Lehr-Lernsettings operationalisieren lassen, so stellen sie dennoch „[...] big-picture aspects“ (Werbach u. Hunter, 2012, S. 78) dar, die gleichzeitig als abstrakte Ziele von derartigen Settings angesehen werden (Kruse et al., 2014). Schließlich sind viele für Spiele typische Eigenschaften auch Eigenschaften guter Lernumgebungen (Prensky, 2001). Im Folgenden soll kurz aufgezeigt werden, wie die Integration beider Domänen in Bartel et al. (2015) gesehen wird und welche Anknüpfungspunkte von Prinzipien kompetenzorientierter Lehre für die Dynamics Constraints, Emotions, Narrative, Progression und Relationships ebenso Gültigkeit besitzen.

**CONSTRAINTS** Constraints sorgen dafür, dass Entscheidungen, die von Spielern innerhalb eines spielifizierten Kontextes getroffen werden, folgenreich sind (Werbach u. Hunter, 2015, S. 27). Sie zeigen damit Handlungsalternativen auf, die sich auf den Spielfluss mittelbar oder unmittelbar auswirken. Zudem schaffen sie eine Balance zwischen Limitierungen und Freiheit für Spieler (Bartel et al., 2015, S. 460).

In einem kompetenzorientierten Lehr-Lernkontext existieren ebenso viele verschiedene Arten von Constraints, auf unterschiedlichen Ebenen, wie beispielsweise die Bearbeitungszeit, die Erfüllung einer Mindestqualitätsanforderung bei der Beantwortung einer Aufgabe oder das Erreichen von bestimmten Lernzielen in einer mehr oder weniger frei wählbaren Reihenfolge. Solche Constraints sind in den meisten Fällen nicht isoliert voneinander zu betrachten, sondern stehen in Wechselwirkung zueinander. Beispielsweise kann hier das bidirektionale Wirken<sup>18</sup> von Qualität und Quantität bei der Bearbeitung von Lernaufgaben herangezogen werden (siehe hierzu z.B. Schneider u. Kreuz (1979)). Wie sich die Constraints in konkreten spielifizier-

<sup>18</sup> Siehe beispielsweise die auf Problem-Based Learning basierende Untersuchung zu Lernzielorientierung bei Studierenden von Verkoeijen et al. (2006).

ten Lehr-Lernsettings ausgestalten bedarf einer sorgfältigen Definition und schafft nicht nur für Lernende einen häufig regelbasierten Orientierungsrahmen, sondern erlaubt ein auf persönlichen Interessen und Entscheidungen basierendes Explorieren<sup>19</sup> von Lernsettings (Kapp, 2012a, S. 66).

**EMOTIONS** Werbach u. Hunter postulieren, dass „Games are powerful and engaging because they tap into the emotions“ (Werbach u. Hunter, 2015, S. 28). Sie zielen also darauf ab, dass spielifizierte Kontexte bei Spielenden eine Form der emotionalen Erregung hervorrufen sollen, welche positiv wie auch negativ sein kann und sich je nach Erregungsart beispielsweise unterschiedlich stark auf die in erster Linie intrinsische Motivation auswirken kann (Werbach u. Hunter, 2015).

Obwohl sowohl für extrinsische als auch intrinsische Motivation gezeigt werden konnte, dass beide die Performanz beim Lernen erhöhen (Cerasoli et al., 2014), so wird angenommen, dass nur letztere zu einer Steigerung der Learning Outcomes beiträgt (Ryan u. Deci, 2000a). Die Spielifizierung von Lehr-Lernsettings kann dazu beitragen, diese notwendige Motivation zu erzeugen, auch wenn individuelle Unterschiede darin bestehen können, wie sich einzelne Game Design-Elemente auf die jeweilige Motivation bei Lernenden auswirken können (Hamari et al., 2014). Die Notwendigkeit einer „motivationalen Imprägnierung“ (Erpenbeck, 2014, S. 21) von Lernprozessen, wie es der Kompetenzforscher Erpenbeck bezeichnet, scheint jedoch unabdingbar, zumal jede Lernhandlung zwingend Emotionen, wie Motivation, voraussetzt und impliziert (Hascher, 2010, S. 13; Erpenbeck, 2014, S. 21). Folglich bedarf es ausgiebiger Überlegungen, welche Emotionen im Zuge einer spielifizierten Intervention geweckt und angesprochen werden sollen. Hierzu existieren zwar einige Studien bzw. Meta-Studien (siehe z.B. Mekler et al. 2013a; Mekler et al. 2013b; Seaborn u. Fels 2015), die ein Mapping von Game Design-Elementen auf eine oder mehrere Emotionen, vor allem Motivationsarten, elaborierten, jedoch steckt dieses Forschungsfeld noch in den Kinderschuhen (Seaborn u. Fels, 2015). Nicht zuletzt ist dafür eine noch nicht vorhandene Standardisierung der Messmethoden verantwortlich (Deterding, 2015).

**NARRATIVE** Narrative (zu deutsch: Narrativ) manifestiert sich in spielifizierten Kontexten durch eine entweder explizite oder implizite Storyline, welche einer inneren konsistenten Logik folgt und in einem bestimmten Kontext existiert (Bartel et al., 2015, S. 460). Der Kontext wird mitunter durch Cons-

<sup>19</sup> Diese Art des Lernens korrespondiert stark mit einer konstruktivistischen Sicht auf Lernen und wird im Sinne von Reich (2012) verstanden.

traints definiert. Erzählungen können, gemäß den Prinzipien kompetenzorientierter Lehre, dafür verwendet werden, um das Lernen strukturell zu unterstützen und Lernenden Klarheit darüber geben, wo sie sich in Bezug auf die Erreichung eines Lernziels befinden (Kapp, 2012a, S. 68). Zudem sollte eine Erzählung nicht ausschließlich linear verlaufen (Kapp, 2012a, S. 67). Gemäß den persönlichen Vorlieben und dem Vorwissen entsprechend, entstehen individuelle Lernpfade, welche regelbasiert, beispielsweise anhand von Quests und darin eingebettete einzelne Tasks bzw. Aufgaben, durchlaufen werden (Bartel et al., 2015). Eine mögliche Unterscheidung der Struktur von Erzählungen ist die in lineare und partiell-lineare Erzählstränge<sup>20</sup>, welche gerichteten und ungerichteten Graphen entsprechen (Bartel u. Hagel, 2016a).

**PROGRESSION** Progression (zu deutsch: Fortschritt) ist die Grundlage für erfolgreiche Kompetenzentwicklung und damit die Ausnutzung von Kompetenzentwicklungspotenzialen. Indikatoren für diese Entwicklung können auf Seite der Kompetenzorientierung kognitive Kompetenzmodelle<sup>21</sup> sein, wie das, welches im Projekt EVELIN entwickelt wurde (Claren u. Sedelmaier, 2012). Die darin enthaltenen Kompetenzniveaus werden durch Lernziele operationalisiert. Wie in Bartel et al. (2015) beschrieben, können die einzelnen Kompetenzniveaus (oder feingranularere Lernziele) auf spielifizierte Zielmetriken<sup>22</sup>, wie beispielsweise Punkte, Levels oder Leaderboards, übertragen und damit abgebildet werden. Unterschieden werden müssen dabei Zielmetriken, die sich auf unterschiedlichen Granularitätsstufen befinden und für Lernende eine Form des *positiv-informationalen Performanz-Feedbacks* (Przybylski et al., 2010) darstellen, um deren Bedürfnis nach Kompetenzerleben zu bedienen (Francisco-Aparicio et al., 2013). So können beispielsweise Components wie Punkte dazu dienlich sein, einen Mikrofortschritt anzuzeigen wohingegen Level, welche sich durch gewisse Punktstaffellungen ergeben, einen Makrofortschritt indizieren (Bartel u. Hagel, 2016b, S. 75). Auch sind wie das Anwendungsbeispiel von Lorenz u. Meier (2014) zeigt, nicht rein quantitative Components, wie Badges, zur Kompetenzdokumentation dienlich. Derartige

<sup>20</sup> Weitere Anordnungsalternativen sind unter anderem zu finden in Schell (2015). Eine Übersicht ist ebenso in Dormans (2012) zu finden. Die dort beschriebenen Anordnungen beziehen sich in erster Linie jedoch auf (Serious) Games und sind deshalb für Gamification in konstruktivistischen Lehr-Lernkontexten nur bedingt übertragbar.

<sup>21</sup> Kompetenzrahmenmodelle (oder auch bekannt als Kompetenzmodelle) existieren in einer großen Vielzahl (siehe z.B. Weinert 2001; Arbeitskreis Deutscher Qualifikationsrahmen 2203). Jedoch wurde das durch Claren u. Sedelmaier (2012) im Rahmen von EVELIN entwickelte Modell bereits in einem spielerischen (mobilen) Lernsetting in Bartel et al. (2014a) verwendet und dokumentiert.

<sup>22</sup> Der Begriff der „goal metrics“ wurde für die Components Punkte, Level und Leaderboard von Zagal et al. (2005) verwendet.

Component-Abhängigkeiten (quantitativ zu quantitativ oder quantitativ zu qualitativ) sind vielfach zu finden und sind je nach pädagogischem Ziel unterschiedlich einsetzbar und für Lernende individuell effektiv (Seaborn u. Fels, 2015).

**RELATIONSHIPS** Relationships (zu deutsch: Beziehungen) stellen in beiden Domänen einen zentralen Bestandteil dar. Sowohl die kompetenzorientierte als auch die spielifizierte Sicht messen sozialen Beziehungen bezogen auf einen Lernkontext eine zentrale Bedeutung zu (Krapp, 1999). Eine mögliche Begründung dafür liefert die Self-Determination Theory (SDT) nach Deci u. Ryan, welche das menschliche Bedürfnis nach sozialer Eingebundenheit als ein universelles Bedürfnis ansieht, das fortwährender Befriedigung bedarf (Deci u. Ryan, 1985; Deci u. Ryan, 1993). Zudem trägt Lernen in sozialen Kontexten und durch Spielifizierung gestützt, dazu bei, individuell-soziale Kompetenzen zu fördern (Giannetto et al., 2013; Hamari u. Koivisto, 2013; Stöcklin et al., 2015).

Betrachtet man die einzelnen Dynamics und deren Anknüpfungsmöglichkeiten an kompetenzorientiertes Lernen, so wird deutlich, dass sich zwei Typen von Dynamics und deren Operationalisierungen herauskristallisieren: Solche, die einen direkten und unmittelbaren verhaltensbeeinflussenden Einfluss auf das Lernen selbst haben (z.B. weil sie als Feedback-Instrumente eingesetzt werden) und jene, die einen deutlich organisatorischen Einschlag besitzen und damit in erster Linie längerfristig und übergreifender wirken (z.B. weil diese vornehmlich den Rahmen für das Lernen vorgeben). Diese Unterscheidung ist nicht zuletzt dadurch begründbar, dass beide Typen von Dynamics auf teilweise sehr unterschiedlichen Granularitätsstufen und damit verschiedenen ausgeprägten Gültigkeitsbereichen agieren. So kann die Belohnung eines Lernenden mit einem Badge als ein unmittelbar konkretes Ereignis charakterisiert werden, während die Festlegung, dass eine bestimmte Anzahl an Badges zu einem neuen Badge führt, ein Ziel darstellt, welches deutlich höher angesiedelt ist und von weit mehr Faktoren abhängig ist, als einer konkreten Aktion durch einen Lernenden. Eine solche Sichtweise ähnelt der von Kapp et al. (2014), welcher eine Unterscheidung zwischen inhaltlicher und struktureller Spielifizierung trifft, jedoch dabei den Wirkungsaspekt auf den Lernenden gänzlich außen vor lässt.

Die getroffenen Überlegungen, sollen als Grundlage für die spätere Umsetzung der theoriegetriebenen Annahmen dienen und darüber hinaus mit in die

technische Umsetzung zur systemunterstützten Erstellung von spielifizierten Lehr-Lernarrangements einfließen.

Die Idee, eine pädagogisch-didaktische mit einer spielifizierten Sichtweise zu verbinden, wurde in der Wissenschaft nicht nur von Bartel et al. (2015) fokussiert. Eine verwandte Arbeit erschien im selben Zeitraum durch Landers (2014). In der darin beschriebenen „theory of gamified learning“ (Landers, 2014) unterscheidet Landers zwischen zwei Prozessen, die beim spielifizierten Lernen auftreten: dem Einfluss von spielifizierten Elementen auf die Beziehung von didaktischem Design und Learning Outcomes, welchen er als einen moderierenden (engl.: moderating) Prozess benennt, sowie den Einfluss, den spielifizierte Elemente direkt auf das Lernen haben, was er als einen vermittelnden (engl.: mediating) Prozess beschreibt (Landers, 2014, S. 9). Landers konzentriert sich in seiner Arbeit jedoch lediglich auf rein fachliche Learning Outcomes und lässt Fragen zu einer holistischeren Sichtweise<sup>23</sup> der Kompetenzorientierung, wie sie in Bartel et al. (2015) vorgestellt wurde, offen. Grundsätzlich jedoch ist eine Übereinstimmung zwischen beiden Arbeiten dahingehend zu erkennen, dass eine gezieltere Ausdifferenzierung der Art und Weise, wie spielifizierte Lehr-Lernarrangements erstellt und deren *Erfolg* beurteilt wird, vonnöten ist und momentane Forschung dies noch zu wenig berücksichtigt. Auch die Typenbildung bei Game Design-Elementen ist ähnlich, wenn auch in beiden Arbeiten nicht vom selben Grundverständnis bezüglich Game Design-Elementen ausgegangen wird. Die Ähnlichkeit kommt diesbezüglich vor allen Dingen dadurch zustande, dass beide Arbeiten eine Typisierung von Game Design-Elementen vornehmen, welche eine Unterteilung in eher steuernde (auf Verhalten abzielende) und inhaltlich (auf Einstellung abzielende) fokussierte Elemente vornehmen. Abstrakter gesprochen, beschreiben beide Arbeiten Möglichkeiten, eine bestimmte Sichtweise auf die in der Taxonomie von Werbach u. Hunter vorgeschlagenen Mechanics vorzunehmen - denn diese werden somit als eine Art Spielmodi interpretiert, welche je nach pädagogischer Zielsetzung durch darüber liegende Dynamics differenziert ausgestaltet und ausgewogen werden. Auf diese Interpretation soll im Folgenden näher eingegangen werden.

---

23 In einer späteren Arbeit beziehen Landers et al. die entwickelte Theorie auf ausgewählte psychologische Theorien und entwickeln die bis dato vorhandene Sichtweise diesbezüglich weiter (Landers et al., 2015). Einige Zeit später erfolgte zudem ein Operationalisierungsvorschlag, welcher Game Design-Elemente mit konkreten, auf der Theorie basierten, Handlungsempfehlungen verknüpft (Landers et al., 2017).

### 2.5.3 Validierung der Theorie

Ebenso wie Landers u. Landers die „theory of gamified learning“ durch ein Lehrexperiment validierten (Landers u. Landers, 2014), wurde die oben vorgeschlagene theoretische Vereinbarkeit von beiden Domänen experimentell validiert (Bartel u. Hagel, 2016b). Das Ziel des Lehrexperiments war herauszufinden, inwiefern sich die vorgeschlagene Verknüpfung auf Basis der Dynamics in Lernsituationen anwenden lässt und welche Möglichkeiten der Operationalisierung der Dynamics, hin zu Mechanics und Components existieren.

#### 2.5.3.1 Der Gamified Course Design Process

In einem ersten Schritt dieses top-down-Ansatzes wurde ein Prozess entwickelt, der das Vorgehen beim Design von kompetenzorientierter und spielifizierter Lehr-Lernarrangements leitet und unterstützt (Bartel u. Hagel, 2016b). Der Gamified Course Design Process (GCDP) (siehe Abbildung 13) geht davon aus, dass ein entsprechendes Kompetenzprofil, wie das bereits erwähnte des BMBF-Projekts EVELIN (Claren u. Sedelmaier, 2012), vorliegt und darin Lernziele sowohl fachlicher als auch überfachlicher Art konkretisiert sind. Der Prozess legt zudem die sechs Lernstufen nach Roth (1983) zu Grunde, die beim Design derartiger Lehr-Lernarrangements Berücksichtigung finden sollen. Roth unterteilt den Lernprozess aus Sicht von Lernenden in die sechs Stufen (1) Motivation, (2) Schwierigkeiten, (3) Lösung, (4) Tun und Ausführen, (5) Behalten und Einüben, sowie (6) Bereitstellen, Übertragung, Integration des Gelernten (Roth, 1983, S. 208ff.). Obwohl ursprünglich für den schulischen Bereich entwickelt, finden sich die Lernstufen in seminaristischen Hochschullehrveranstaltungen, welche an der Arbeit mit authentischen Problemstellungen ausgerichtet sind und damit einer konstruktivistischen Sichtweise auf Lernen folgen, wieder. Diese Lernstufen geben Lehrenden in der Rolle der Designer von Lehr-Lernarrangements ein empathisches Werkzeug dafür, wie der Lernprozess bei Lernenden ausgestaltet sein kann und welche Schritte im Einzelnen dafür erfolgen sollten.

Auf Basis der Lernziele (learning outcomes) und abhängig von den darüber liegenden Kompetenzen werden Erstere in einzelne Lernaktivitäten bzw. Lehr-Lernarrangements zusammengefasst bzw. konsolidiert. Dabei kommen in einem Arrangement eine oder mehrere didaktische Methoden zur Anwendung. Wichtig ist dabei, dass die Auswahl einer Methode nicht unabhängig von den Game Design-Elementen erfolgen kann und deshalb eine wechselseitige Beziehung zwischen beiden Elementen besteht (Bartel u. Hagel, 2016b).

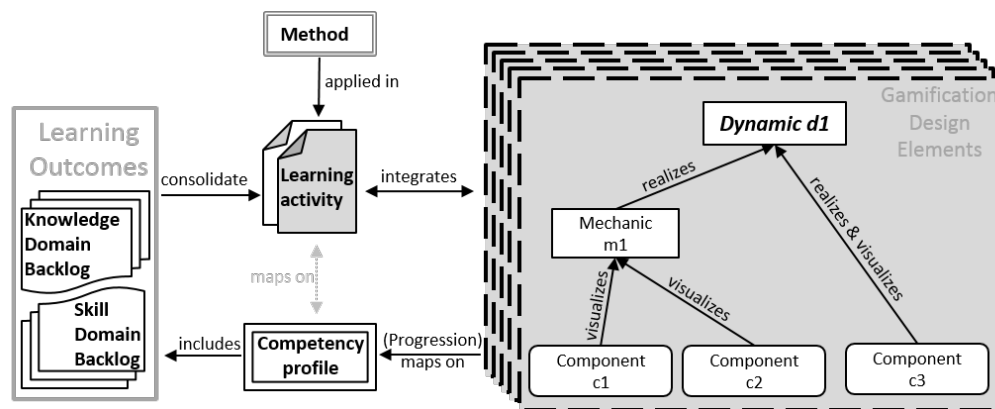


Abbildung 13: Der GCDP von Bartel u. Hagel (2016b, S. 75)

So würde sich beispielsweise eine *Briefmethode*<sup>24</sup> (Reich, 2012) nur bedingt für den Einsatz von sozialen Game Design-Elementen eignen, da die klassische Methodenausprägung keine soziale Interaktion zwischen den Lernenden, sondern nur zu Lehrenden vorsieht. Natürlich ließe sich diese Festlegung entsprechend modifizieren, würde jedoch die Methode verändern. Im Umkehrschluss würde sich eine Methode wie ein *Rollenspiel* (Reich, 2012) auch nur dann eignen, wenn aus einer spielifizierten Sichtweise ein Rollenkonzept vorgesehen ist und ein Einsatz von Rollen inhaltlich-didaktisch sinnvoll verortet werden kann. Ebenso sind beidseitige und nicht hierarchisierte Wechselwirkungen zwischen Lerninhalten und Methode(n) (Meyer, 2016), als auch zwischen Methoden und Game Design-Elementen unter Berücksichtigung von Lerninhalten zu beachten (Bartel u. Hagel, 2016b).

Einhergehend mit der vorläufigen Festlegung auf Methode(n), Lerninhalt(e) und Game Design-Elementen, erfolgt die strukturierte Ausarbeitung letzterer, beginnend bei den Dynamics. Grundsätzlich werden von Bartel u. Hagel zwei Wege gesehen, wie sich Dynamics operationalisieren lassen (Bartel u. Hagel, 2016b, S. 75):

- Eine Dynamic wird durch eine Mechanic<sup>25</sup> umgesetzt, wobei die Mechanic für Lernende durch eine Component sichtbar wird.

Beispiel: Die Dynamic *Emotion* wird durch die Mechanic *Competition*

<sup>24</sup> Bei dieser Methode werden Lernende aufgefordert, eine Problemstellung einer erdachten Person in Form eines durch Lernende zu verfassenden Briefes zu lösen. Die Formulierung des Briefes geschieht empathisch und unter Nutzung des eigenen Wissens sowie des Wissens der erdachten Person (Reich, 2012).

<sup>25</sup> An dieser Stelle wird bewusst die Terminologie von Werbach u. Hunter (2015) verwendet, um ein unterschiedliches Verständnis der Elementklassen dieser Taxonomie auszuschließen.



(*Wettbewerb*) erzeugt, welche für Lernende in Form eines *Leaderboards* sichtbar wird.

- Eine Dynamic kann ebenso auch direkt durch eine Component realisiert und damit für Lernende sichtbar gemacht werden.

Beispielhaft kann hier die Beziehung von der Dynamic *Progression* zu den Components *Points* angeführt werden.

Beide Ansätze erfordern zweifellos ein gewisses Maß an Kreativität und Erfahrung, um die Verbindung von rein pädagogischen Inhalten und Zielen mit geeigneter Spielifizierung *erfolgreich* zu erstellen. Hilfestellungen können bei diesem Definitionsprozess die in (Bartel u. Hagel, 2016b) vorgeschlagenen<sup>26</sup> inhaltlichen Elemente für Dynamics sein, wie in Tabelle 2 zu finden. Die einzelnen Dynamics und deren inhaltlichen Beschreibungselemente sind als jeweilig eine Perspektive zu verstehen, aus deren Summe sich ein kompetenzorientiertes und spielifiziertes Lehr-Lernarrangement spezifizieren lässt.

Dynamic	Inhaltliche Beschreibungselemente
Constraints	Regeln, Zeit, Umgebungsbedingungen, verfügbare bzw. benötigte Ressourcen, Belohnungssystem(e)
Emotions	Spaß, Wettbewerb bzw. Rivalität, Überraschungen, Neugierde, Kohärenzwahrnehmung, Kompetenz- und Autonomieerleben
Narrative	Aktivator/Trigger bzw. Motivator, Storyline, Szenario, Quests
Progression	Quantitative (z.B. Points, Levels, Leaderboards) oder qualitative (z.B. Badges, virtual Goods) Zielmetriken
Relationships	Sozialform der Bearbeitung, Interaktionswege und -häufigkeit, Interaktivitätserwartung

Tabelle 2: Inhaltliche Beschreibungselemente von Dynamics nach Bartel u. Hagel (2016b, S. 75)

Ein besonderes Augenmerk wird bei dem vorliegenden Prozess auf die Dynamic Progression gelegt. Wie bereits ausgeführt, bedarf es einer Form von quantitativer (z.B. Points) oder qualitativer (z.B. Badges) Zielmetrik, um für

<sup>26</sup> Die Aufzählung ist nicht vollständig, sondern soll beispielhaft darstellen, welche Beschreibungselemente von entsprechenden Dynamics fokussiert werden können. Weitere Elemente sind darüber hinaus möglich und sicherlich auch sinnvoll.

Lernende deren gezeigte Performanz zu beurteilen. Beide Typen von Fortschrittsmetriken können jedoch auf die jeweiligen Kompetenzniveaus eines Kompetenzprofils und damit dem Erreichen von bestimmten Lernzielen abgebildet werden. Jedoch gilt dieses Mapping in erster Linie den fachlichen Kompetenzen. Überfachliche Kompetenzen können in erster Linie durch qualitative Metriken beurteilt werden (siehe z.B. Seaborn et al. (2013)), welche beispielsweise als Ergebnis eines Peer-Votums an Lernende vergeben werden (Bartel u. Hagel, 2016b, S. 76).

Folglich lässt sich also das Durchlaufen des GCDP zum Design von spielfizierten und kompetenzorientierten Lehr-Lernarrangements anhand der folgenden 7 Schritte zusammenfassen (Bartel u. Hagel, 2016b, S. 76):

1. Auswählen oder Definieren von für das Vorhaben relevante Kompetenzen (z.B. basierend auf einem vorhandenen Kompetenzprofil).
2. Basierend auf den ausgewählten Kompetenzen werden die damit verbundenen Lernziele definiert (z.B. unter Zuhilfenahme von Inhaltstemplates (Figas et al., 2014)).
3. Unter Berücksichtigung der Zielgruppe und der einzusetzenden Spielfizierung werden für das Lehr-Lernarrangement eine oder mehrere didaktische Methoden ausgewählt und bei Bedarf angepasst.
4. Das Lehr-Lernarrangement wird zunächst allgemein beschrieben.
5. Alle einzusetzenden Dynamics werden bezüglich ihrer Beziehung zu Mechanics oder Components beschrieben und wie sie sich in das Lehr-Lernarrangement integrieren. Dabei ist ein besonderer Fokus zu legen auf...
  - ... die einzelnen Schritte, die zum Durchlaufen des Lehr-Lernarrangements notwendig sind (vorher, währenddessen und nachher).
  - ... die Feedbackmechanismen für Lernende.
6. Die einzusetzenden Methoden werden innerhalb des Lehr-Lernarrangements entsprechend den einzusetzenden Game Design-Elementen angepasst. Optional kann auch eine inhaltliche Veränderung der Lernmaterialien erfolgen (inhaltliche Spielfizierung).
7. Fortschrittsmetriken werden auf die jeweiligen Kompetenzniveaus und damit verbundene Lernziele gemappt.

Neben dem vorgestellten GCDP existieren noch eine Vielzahl an weiteren (Sub-)Prozessen, die für das Design von spielifizierten Anwendungen herangezogen werden können. Auch handelt es sich bei dem GCDP um einen vergleichsweise generischen Prozess, der zunächst die Vergleichbarkeit von verschiedenen spielifizierten und kompetenzorientierten Lehr-Lernarrangements erhöhen soll. Im Sinne der Artefaktentwicklung des DSR können nun einzelne Teile des Prozesses weiter präzisiert und validiert werden. So findet eine explizite Zielgruppenanalyse, wie sie von Morschheuser et al. (2017) vorgeschlagen werden, aktuell noch keine Anwendung im GCDP. Auch Gamification Design Patterns, wie von Ašeriškis u. Damaševičius (2014) erarbeitet, könnten dem GCDP die durch DSR geforderte methodische Strenge weiter erhöhen. Darüber hinaus bietet das von Deterding (2015) bereits auf Gamification adaptierte Werkzeug der *Design Lenses* eine weitere Möglichkeit, unterschiedliche Perspektiven beim Design von spielifizierten und kompetenzorientierten Anwendungen einzunehmen und damit rückkoppelnd den GCDP systematisch zu erweitern. Diese Erweiterungsmöglichkeiten sind nur beispielhaft angeführt, um zu zeigen, dass die feingranularere Ausgestaltung des GCDP durchaus möglich ist.

Zunächst soll jedoch eine ebenso durch den DSR geforderte Evaluierung des GCDP erfolgen. Sie soll zeigen, inwiefern sich eine durch den GCDP gestützte Erstellung von spielifizierten und kompetenzorientierten Lehr-Lernarrangements verwirklichen und schließlich in die Lehre integrieren lässt und damit die theoretische Vereinbarkeit von kompetenzorientierter Lehre und Spielifizierung praktisch festigt.

### 2.5.3.2 *Der Gamified Course Design Record*

Hierfür wird der GCDP instanziiert und gemäß dem vorgestellten Sieben-Schritte Plan durchlaufen. Die Instanziierung des GCDP wird als Gamified Course Design Record (GCDR)<sup>27</sup> bezeichnet (Bartel u. Hagel, 2016b). Die Beziehung zwischen GCDP und GCDR ist vergleichbar mit aus der Objektorientierung bekannten Beziehung von Klasse zu Objekt.

Es folgt eine retrospektive Beschreibung des Lehr-Lernarrangements von Bartel u. Hagel (2016b). Zunächst erfolgte dort eine Auswahl der relevanten Kompetenzen, der damit konnotierten Lernziele und damit der Lerninhalte. Das Lehr-Lernarrangement hatte den Zweck eine in der Vergangenheit durch den Dozenten subjektiv empfundene Hürde beim Erlernen des Command Proces-

---

<sup>27</sup> Der vollständige tabellarische GCDR befindet sich im Anhang A.1 dieser Arbeit.

sor mit Undo Design Patterns (Geirhos, 2015, S. 222ff.; Buschmann et al., 1996, S. 277ff.) zu überwinden. Diese Hürde bestand nicht nur durch das Verständnis des Patterns, sondern auch durch dessen korrekte Anwendung. Demnach wurde das Lernziel definiert, welches neben den Kompetenzniveaus Erinnern, Verstehen, Erklären auch das Kompetenzniveau Verwenden vorsieht und damit studierendenzentriert formuliert: Nach der Teilnahme an der Lerneinheit, sind Studierende in der Lage, das Command Processor Design Pattern mit undo, unter Anleitung, in für sie neuen Situationen zu verwenden (Bartel u. Hagel, 2016b, S. 76). Analog wurden die Lernziele auch für darunterliegende Kompetenzniveaus entsprechend ausformuliert. Zuzüglich zu den rein fachlichen Kompetenzen, wurden ebenso Lernziele aus den überfachlichen Kompetenzen Problemlösefähigkeit, Kommunikationsfähigkeit und Präsentationsfähigkeit abgeleitet. Die folgenden drei überfachlichen Lernziele spiegeln dies wider (Bartel u. Hagel, 2016b, S. 76):

Nach der Teilnahme an der Lerneinheit, ...

- ... sind Studierende in der Lage, neue und zielführende Ideen anhand einer inhaltlich verwandten Problemstellung zu entwickeln.
- ... haben Studierende Erfahrungen entwickelt, wie in Gruppen kommuniziert und kooperiert wird, um gemeinsam ein Problem zu lösen.
- ... sind Studierende in der Lage, Ergebnisse einer Gruppe vor Kommilitonen zu präsentieren.

Methodisch wurden für das Lehr-Lernarrangement situiertes Lernen<sup>28</sup> (Lave u. Wenger, 1990) in Kombination mit Gruppenarbeit (Klippert, 2009) ausgewählt. Ein *Tailoring* der beiden Methoden war hinsichtlich der Dynamics vonnöten und erfolgte in Wechselwirkung zu der Operationalisierung der Dynamics.

### *Narrative*

Um den inhaltlichen Kontext weiter auszuarbeiten, wurde zunächst die Dynamic Narrative definiert und damit relevante Meilensteine im Ablauf des Arrangements festgelegt. Gemäß den Lernstufen nach Roth (1983) bedarf es zu Beginn einer initialen Motivation bzw. eines Triggers, welcher den Lernwunsch

<sup>28</sup> Beim situierten Lernen handelt es sich nach Lave u. Wenger eher um eine Theorie, als eine konkrete Methode mit Handlungsempfehlungen zum Ablauf. Jedoch definieren die Autoren Kriterien, anhand deren ihre Vorstellungen von situiertem Lernen verdeutlicht werden. Sie gehen von der Grundannahme aus, dass Wissen in authentischen Situationen verortet sein muss und bei dessen Aneignung soziale Interaktion in *communities of practice* eine zentrale Rolle spielt (Lave u. Wenger, 1990).

bei Lernenden weckt und damit den Lernprozess initiiert. Dies wurde in Form einer Email<sup>29</sup> von einer für die Lernenden bekannten Firma erreicht, welche den Lernenden ausgehändigt wurde. Die Email enthielt eine authentische Arbeitsbeschreibung, welche die Lernenden bat, eine kreative und passgenaue Lösung zu entwickeln, wie die Firma das Command Processor Pattern einem Kunden ihrerseits vermitteln kann, welcher lediglich grundlegendes Wissen in objektorientiertem Design besitzt. Die Art der Lösung, also ob eine Präsentation, ein Poster oder beispielsweise Handouts dieses Ziel erreichen sollten, war nicht festgelegt. Hier wurde eine erste Operationalisierung getroffen, in dem die konkrete Aufgabe insgesamt drei Quests (inkl. Teiltasks) enthielt, welche von Lernenden in einer Beraterrolle durchlaufen werden sollten (Bartel u. Hagel, 2016b, S. 76): Der erste Quest *Hilfe für die Firma* war erfüllt, wenn der Kontext angenommen, das bereits vorhandene Wissen zu diesem Thema in Diskussion erfasst und erste kreative Lösungsideen geäußert wurden (Lernstufe 2 bzw. 3 nach Roth (1983)). Überprüft wurde die Erreichung durch inhaltliche Fragen an die jeweiligen Gruppen durch den Dozenten. Der zweite Quest *Design der Lösung* bestand in der gemeinsamen Findung eines adäquaten Präsentationsformats und der Ausarbeitung einer Lösung gemäß den Anforderungen. Erfüllt wurde dieser Quest, wenn eine Gruppe einen für sie fertigen Lösungsvorschlag vorweisen konnte (Lernstufe 4 nach Roth (1983)). Der dritte und letzte Quest *Präsentation und Votieren* umfasste die gruppenweise Präsentation der erarbeiteten Lösungen und ein peer voting, welches die Gruppe mit dem *besten* Erklärungsmaterial bestimmen sollte, welches schließlich der Firma zur Verfügung gestellt werden sollte (Lernstufe 5 und 6 nach Roth (1983)).

### *Emotions*

Neugierde und das Gefühl von Relevanz des eigenen Tuns sollte von der ausgehändigten Email bei den Lernenden geweckt werden. Der Wettbewerbsgedanke wurde dadurch erzeugt, dass nur die *beste* Lösung an die Firma gesendet werden konnte, wobei sich die einzelnen Gruppen aus Lernenden gegenseitig bewerteten. Somit kristallisierten sich die folgenden drei Mechanics heraus, welche durch Components weiter operationalisiert wurden (Bartel u. Hagel, 2016b, S. 77):

- Mechanic<sub>1</sub> *Competition*: Die Lernenden durchlaufen gemeinsam in Gruppen die Ihnen gestellten Quests. Dabei treten die einzelnen Gruppen in

<sup>29</sup> Die Email wurde im Vorfeld auf Wunsch des Autors von der Firma erstellt und war demnach fingiert, was zum Zeitpunkt der Durchführung des Lehr-Lernarrangements den Lernenden nicht bekannt war.

einem Wettbewerb gegeneinander an, denn die Gruppe mit dem höchsten Ergebnis beim peer-voting, als Abschluss von *Quest* drei, gewinnt. Zur Visualisierung des peer-votings werden *Points* eingesetzt, welche das Gruppenergebnis einer jeden Gruppe in einem *Leaderboard* zeigen.

- *Mechanic<sub>2</sub> Cooperation*: Zur Erfüllung der *Quests* arbeiten die Lernenden in Kleingruppen (3 bis 5 Personen), welche auf ein hohes Maß an Kommunikation und Kooperation angewiesen sind. Gemäß ihren Kenntnissen und Fähigkeiten müssen zur Questerfüllung unterschiedliche Rollen innerhalb der Gruppen besetzt werden. Aus Gesamtsicht befinden sich jedoch alle Lernenden in der Rolle als Berater, bezogen auf die auftragserteilende Firma.
- *Mechanic<sub>3</sub> Challenges*: Die Lernenden werden durch die ausgehändigte Email und die darin enthaltenen *Quests* herausgefordert, ihre Kompetenzen zu demonstrieren und dabei an vorhandenes Wissen anzuknüpfen. Vorausgehend ist die selbstreflexive Beurteilung der vorhandenen Kompetenzen im Bezug auf die gestellten Herausforderungen durch die Lernenden.

### *Relationsships*

Bezogen auf die *Dynamic Relationsships* und durch die Einbettung der Methode Gruppenarbeit wurden die Lernenden dazu aufgefordert in Teams zu arbeiten. Dadurch wird unter anderem das Gefühl von sozialer Eingebundenheit erzeugt, ein – nach der SDT – Grundbedürfnis von Menschen (Deci u. Ryan, 1985). Der Grad an Gruppen-interner Abstimmung wurde in keinsten Weise vorgegeben, ebenso wenig, wie die Zusammensetzung der einzelnen Teams.

### *Progression*

Progression wurde anhand der *Mechanic Feedback* und der *Component Quests* und darauf beziehend, mit Hilfe der quantitativen Zielmetriken *Points* und *Levels* für die Lernenden sichtbar gemacht – *Mechanics* bzw. *Components* die nach Kapp sich sehr gut als Feedback Mechanismen in spielifizierten Kontexten eignen (Kapp, 2012a, S. 68). Hierzu wurde den Lernenden ein strukturiertes Feedback gegeben, welches einerseits während der einzelnen *Quests* durch das Team informell beim Dozenten eingefordert werden konnte und andererseits als Bestandteil einer erfolgreichen Questerfüllung in Form von *Levels* gepaart mit kurzem informellen Feedback den Gruppen mitgeteilt wurde.

Das Feedback nach Questerfüllung folgt damit der Argumentation von Kapp et al.. Er attestiert dem *Debriefing*, also dem reflexiven und durch Feedback-getriebenen Aufgreifen des Erlernten, nach jeder Lerneinheit eine hohe Wichtigkeit in Bezug auf dessen Internalisierung:

„Then provide a debriefing to the learner so he or she can internalize the lessons and transfer those lessons to other experiences. The debriefing provides the chance for generalization of the learning [...]. Without the generalization, the learning is confined only to that one experience. In other words, experience without reflection is just experience“ (Kapp et al., 2014, S. 18).

Beim Feedback wurde das bereits beschriebene Mapping von *Level* zu Kompetenzniveau wie folgt vorgenommen: Die Erfüllung des ersten *Quests Hilfe für die Firma* führte zu einer Steigerung des *Levels* von null auf drei welches dem Kompetenzniveau Erklären entsprach und voraussetzte, dass „[...] Zusammenhänge, Abhängigkeiten und Ähnlichkeiten von Informationen [erkannt] und in eigenen Worten erklärt [werden können]“ (Claren u. Sedelmaier, 2012, S. 648). Es wurde angenommen, dass damit die Kompetenzniveaus Erinnern und Verstehen ebenfalls erreicht wurden, und damit der Argumentation von Claren u. Sedelmaier (2012) gefolgt. Die Erfüllung von dem zweiten *Quest Design der Lösung* führte zu einer weiteren Steigerung des *Levels* von drei auf vier und implizierte damit das Erreichen des Kompetenzniveaus Verwenden (unter Anleitung deshalb, weil Hilfestellungen durch den Dozenten zulässig waren) (Bartel u. Hagel, 2016b, S. 77). Der Abschluss des dritten und letzten *Quests* war nicht an ein Kompetenzniveau gebunden, da es sich um nicht rein fachliche Kompetenzen handelte. Vielmehr äußerte sich die Vollendung durch den Erhalt einer Bewertung von anderen Gruppen anhand eines peer-votings. Die Bewertung erfolgte basierend auf den Kriterien Präsentationsstil, Qualität der Erklärungen, Originalität und Verständlichkeit der vorgestellten Lösung (Bartel u. Hagel, 2016b, S. 77). Dabei wurde jedes Kriterium mit einem Punktwert bewertet (Skala 1 - sehr schlecht bis 5 - sehr gut) und daraus das arithmetische Mittel berechnet, was schließlich die Bewertung einer Gruppenlösung von einer anderen Gruppe ergab. Schlussendlich wurde ebenfalls auf Basis der Einzelbewertungen das arithmetische Mittel angewandt, um zu dem Gesamtdurchschnittspunktwert zu kommen, welcher in einem *Leaderboard*, zusammen mit den Einzelpunktwertungen, festgehalten wurde.

### Constraints

Das Arrangement war an *Constraints* gebunden, welche teilweise durch den Dozenten beeinflusst werden konnten, jedoch teils auch starr von außen vorgegeben wurden. Ein Beispiel für letztere Art von *Constraints* ist die verfügbare Zeit, welche für das gesamte Arrangement zur Verfügung stand und insgesamt 90 Minuten betrug. Zur inhaltlichen Einstimmung durch den Dozenten wurden 10 Minuten Zeit investiert, weitere 45 Minuten standen für die Arbeit an *Quest* eins und zwei zur Verfügung und die verbleibenden 35 Minuten für *Quest* drei. Vor Beginn von *Quest* eins wurde den Lernenden Büromaterial in Form von leeren Plakaten, Stiften und diverses farbiges Papier ausgegeben, dessen Nutzung fakultativ war. Zudem war ein weiterer *Constraint*, jedoch mehr an den Dozenten gerichtet, dass die Lernenden über die fingierte Mail am Ende des Arrangements unterrichtet werden müssen. Aufgrund von räumlicher und personaler Ressourcen musste zudem eine Maximalteilnehmerzahl von 25 Lernenden beachtet werden, was in etwa fünf Gruppen entsprach, die durch einen Lehrenden während des Arrangements gleichzeitig betreut wurden.

Als Mechanics kamen *Win States* zum Einsatz. Der Wettbewerb wurde dann durch eine Gruppe gewonnen, wenn diese den höchsten durchschnittlichen Gruppenpunktwert erreicht hatte. Dies wurde bereits in der Mechanic *Feedback* der *Dynamic Progression* beschrieben. Darüber hinaus wurde als weitere Mechanic *Rewards* eingesetzt. Neben der Belohnung, dass die beste Gruppenlösung an die Firma gesendet wurde, erhielten die Gewinner zudem ein materielles *Achievement* in Form eines Preises. Dieser bestand in einer Tasse für jedes Gruppenmitglied, welche durch die Firma kostenfrei bereitgestellt wurde.

Eine erhöhte Verständlichkeit der Constraints ist gegeben, wenn diese etwas formalisierter formuliert werden können, als eine reine textuelle Beschreibung dies umsetzt. Deshalb bietet sich hier eine Formulierung einer Wenn-Dann-Regel an, welche im Wenn-Clause eine oder mehrere Bedingungen enthält, die wenn sie alle erfüllt sind, dazu führen, dass der Dann-Clause mit all seinen enthaltenen Konsequenzen ausgeführt wird. Diese Form der Darstellung kommt im GCDR ebenfalls zur Anwendung und soll beispielhaft im Folgenden an der Mechanic *Rewards* beschrieben werden:

WENN  $x_{arithm}(Gruppe_n)$  Maximum von allen  $x_{arithm}(Gruppe_n)$  ist UND alle Gruppen einen Punktwert  $> 0$  besitzen UND alle Gruppen *Quest* eins, zwei und drei abgeschlossen haben DANN gib Feedback in Form des Rewards *Einsendung bestes Ergebnis* an Firma UND belohne jedes Gruppenmit-



glied der Gewinnergruppe mit einer Tasse.

Diese Art der Formulierung soll im Hinterkopf behalten werden, für die spätere Entwicklung einer IT-gestützten Lösung zur Erstellung derartiger kompetenzorientierter und spielifizierter Lehr-Lernarrangements.

### 2.5.3.3 Ablauf

Das beschriebene Lehr-Lernarrangement wurde in einer *Software Engineering 1 Übung* abgehalten, welche im zweiten Semester des Bachelorstudiengangs Wirtschaftsinformatik an der Hochschule Kempten verortet war. Das Wissen über das Design Pattern wurde in der dazugehörigen Vorlesung allgemein und theoretisch ohne Anwendungsbeispiel zehn Tage vor dem Stattfinden der Übung vermittelt. An dem Arrangement nahmen freiwillig 15 Studierende teil, welche sich in vier Gruppen zu je drei bzw. vier Lernenden zusammenfanden. Das Arrangement war im zweiten Drittel des Semesterverlaufs platziert. Demnach war dem Dozenten die Gruppe bereits bekannt. Während der Durchführung dokumentierte der Dozent markante Geschehnisse oder Verhaltensweisen der Lernenden stichwortartig und anonym als Teil des GCDP.

Nach der Aushändigung der Email durch den Dozenten und die anschließende Klärung von auftretenden studentischen Fragen sowie dem Vorstellen der *Constraints* und der Inhalte begannen die Lernenden mit der Bearbeitung der *Quests*. Zu Beginn des ersten Quests war eine gewisse Skepsis gegenüber der Echtheit der ausgehändigten Email zu beobachten, welche jedoch schnell verschwand, als die Skeptiker beobachteten, dass sich andere Gruppen schon mitten in der Bearbeitung befanden (Bartel u. Hagel, 2016b, S. 77). Zudem entstand die durch den Dozenten erhoffte Wettbewerbsdynamik, welche sich durch einzelne kompetitive verbale Aussagen von konkurrierenden Gruppen zeigte. Ebenso von Interesse war die Eigendynamik, welche innerhalb der Gruppen entstand. Ausnahmslos jede Gruppe diskutierte zunächst über die notwendigen Fähigkeiten zur Bewältigung der Quests und wer dabei welchen Part gemäß den persönlichen Fähigkeiten übernehmen könne. Die dabei festgestellten Unterschiede im Wissen um dieses Pattern wurden durch internes Coaching versucht auszugleichen, bevor sich dem ersten Quest gewidmet wurde. Während der Bearbeitung von Quest zwei gab es, wie auch bei Quest eins, von studentischer Seite aus nur sehr wenige Rückfragen an den Dozenten. Das entsprechende Feedback nach Abschluss des ersten Quests wurde als Feedback angenommen und die Rückfragen seitens des Dozenten entsprechend erfolgreich beantwortet. Nach Abschluss von Quest zwei folgte die Gruppenpräsentation der erarbeiteten Lösungen. Nicht durch den Dozenten intendiert aber trotzdem erfreulich war die intensive Diskussion über

die Vor- und Nachteile einer Lösung, welche sich nach jeder Präsentation anschloss. Studentische Aussagen wurden gegenseitig kritisch hinterfragt oder Lob für Kreativität von Ergebnissen geäußert. Das anschließende Votieren brachte einen knappen Gruppensieger<sup>30</sup> hervor. Die vorgegebenen Zeiten für die Bearbeitung der Quests wurden weitestgehend eingehalten.

#### 2.5.3.4 Experimentelle Feedback-Studie

Im Anschluss an den Abschluss von Quest drei wurde eine experimentelle Studie durchgeführt, die im Folgenden vorgestellt wird.

##### *Ziele*

Die Studie hatte zum Ziel die theoretisch aufgezeigte Vereinbarkeit von kompetenzorientierter Lehre und Spielifizierung mit Game Design-Elemente praktisch zu validieren, die dazu notwendigen Instrumente (den GCDP und der GCDR) auf deren Anwendbarkeit für Dozierende zu überprüfen und zudem die folgenden beiden Forschungsfragen zu beantworten (Bartel u. Hagel, 2016b, S. 78):

- Haben die Lernenden, nach ihrer persönlichen Auffassung, die durch den Dozenten angestrebten Kompetenzlevel für dieses Thema erreicht?
- Hat sich die Kombination aus den Komponenten *Points*, *Levels* und *Leaderboards* in Verbindung mit der *Dynamic Narrative* positiv auf die Lernmotivation der Studierenden ausgewirkt?

##### *Studiendesign*

Ein vorwiegend qualitativer Feedback-Fragebogen wurde an die Studierenden ausgegeben. Er enthielt je eine Frage zu den oben genannten Forschungsfragen, wobei die jeweilige Beantwortung der Frage eine Begründung vorsah, einen Freitextbereich, in dem Studierende ihre Meinung zu dem Lehr-Lernarrangement allgemein äußern konnten, eine weitere Frage, welche auf Verbesserungen des Arrangements abzielte, sowie die Frage nach einer abschließenden Bewertung des Arrangements (10 - sehr gut und 1 - sehr schlecht). Insgesamt wurden n=13 Feedbacks ausgefüllt und anschließend die Freitextantworten

---

<sup>30</sup> Bei der erarbeiteten Lösung handelte es sich um ein Plakat, welches einen Roboter beim Einkaufen zeigt. Der Roboter hatte die Fähigkeit Lebensmittel in einen Einkaufswagen zu legen und diese wieder zurück in das Regal zu stellen. Das Plakat zeigte eine direkte Verknüpfung des UML-Modells des Patterns zu den Aktionen des Roboters. Ein entsprechendes Bild des Plakats ist in Bartel u. Hagel (2016b, S. 77) zu finden.

ten mit MAXQDA mit Konzepten kodiert und ausgewertet. (Bartel u. Hagel, 2016b)

### *Ergebnisse*

Das Arrangement wurde insgesamt mit einem Gesamtergebnis von 9,46 von 10 möglichen Punkten bewertet. Dieses positive Ergebnis wurde auch in den einzelnen Freitextstatements bestätigt. Alle 13 Bewertenden beantworteten die erste Forschungsfrage positiv, wonach die Lernenden nach ihrem persönlichen Empfinden die angestrebte Kompetenz erreicht hatten. Häufig (n=10 Erwähnungen) wurde in diesem Zusammenhang erwähnt, dass die gegenseitigen Präsentationen und die anschließende Diskussion positiv zum Erreichen des Kompetenzniveaus beitrugen. Diese „ungewohnte Art zu lernen“, so eine teilnehmende Person, hatte zur Folge, dass sich „jeder mit dem Thema auseinandergesetzt und diskutiert hat“. Ebenfalls wurde als positiv bezüglich der Kompetenzentwicklung die „kreative Art zu lernen“ herausgehoben, welche durch die „Praxisorientierung die Motivation zum Lernen erhöht“. Auch der Wettbewerb wurde als positiv und anspornend empfunden (Bartel u. Hagel, 2016b, S. 78).

Von einigen Studierenden (n=4) wurde jedoch die verfügbare Zeit zur Erledigung der Quests kritisiert. Sie wurde als zu gering empfunden und der dadurch entstandene Zeitdruck sei zu hoch gewesen. Jedoch waren diese Äußerungen die einzigen negativen Statements. Deshalb kann die erste Forschungsfrage positiv beantwortet werden.

Bezogen auf die zweite Forschungsfrage konnten keine direkten Erwähnungen der Game Design-Elemente in den Feedbacks ausgemacht werden (Bartel u. Hagel, 2016b, S. 78). Dennoch wurden die Motivationseffekte vielfach positiv geäußert, welche durch den kombinatorischen Einsatz aller Game Design-Elemente erzeugt wurden. So wurde beispielsweise von einer teilnehmenden Person gesagt, dass „die Aufteilung in machbare kleine Quests Strukturen schafft“, welche für die Person „nicht erschlagend wirkt“ und ermöglicht, dass „man Schritt für Schritt vorgeh[t]“ (Bartel u. Hagel, 2016b, S. 78). Unter Berücksichtigung des teilweise fehlenden Datenmaterials zur zweiten Forschungsfrage kann jedoch deren Beantwortung als positiv angenommen werden, wenngleich ein wissenschaftlich eindeutiger Beleg, in Form von direkten Aussagen beziehend auf die Frage, nicht vorgelegt werden kann. Die Annahme beruht damit auf den indirekten aber vielfach vorhandenen Aussagen bezüglich einem gesteigerten Motivationsempfinden seitens der Studierenden und den während des Arrangements dokumentierten Beob-

achtungen des Dozenten. Conor et al. unterstützen diese Erkenntnis und beschreiben es als deutlich zielführender, wenn ein zuvor definiertes Verhalten bei Lernenden direkt beobachtet werden kann, als die Erreichung eines Verhaltensziels über beispielsweise einen Fragebogen abzusichern (Conor et al., 2014, S. 97).

### *Diskussion*

Das Arrangement scheint damit die gesetzten Lernziele erfüllt und die Kompetenzentwicklungspotentiale bei den Studierenden ausgeschöpft zu haben. Nichts desto trotz gibt es einige Dinge, die bei der Betrachtung der vorgestellten Ergebnisse dem äußeren Betrachter präsent sein müssen.

Das alleinige Aufbringen des erforderlichen und hohen Aufwands zum Design und zur Durchführung eines solchen Arrangements macht es per se nicht zu einem für alle Beteiligten erfolgreichen Arrangement. Vielmehr ist es vonnöten, dass Lehrende und Lernende sich ausreichend im Vorfeld kennengelernt haben und ein (Vertrauens-)Verhältnis aufbauen konnten, was an anderer Stelle bereits als wichtiger Einflussfaktor auf den Lernprozess nachgewiesen werden konnte (siehe Arbeiten zu Lehrende-Lernende-Beziehung von z.B. Apperl u. Brenn (1991) oder der viel zitierten Meta-Studie über Einflussfaktoren auf das Lernen von Hattie et al. (2013)). Dieser These folgend stellt diese Beziehung einen nicht unerheblichen Faktor dar, ob ein solches Arrangement gelingen kann. Zudem müssen Lehrende in der Lage sein, die Lernenden, mit denen sie regelmäßig Umgang pflegen, einzuschätzen, ob ein solches Arrangement auf entsprechenden Anklang bei Lernenden trifft. Nachdem die Persönlichkeiten von Lernenden sehr heterogen sein können, bedarf es einer solchen Einschätzung im Zuge der Planung eines derartigen Arrangements. Orientierung, welche Typen von Lernenden aus spielifizierter Sicht identifiziert werden können, können Fragebogen-basierte Spieler-Taxonomien<sup>31</sup> wie die von Bartle (1996) bieten. Sie unterteilen aus Sicht der Spielifizierung Spieler in Typen (in dem Fall von Bartle in Killers, Socialisers, Achievers und Explorers (Bartle, 1996)), welche unterschiedliches synthetisiertes Verhalten im Spiel zeigen, wobei eine Korrelation von realem Persönlichkeitstyp zu virtuel-

31 Die von Bartle (1996) erstellte Taxonomie ist nicht unumstritten, zumal sie für klassisches Game Design entworfen wurde (Bartle, 1996) (es jedoch auch Anwendungen im Gamification Umfeld gibt (Ferro et al., 2013; Sangkyun u. Ko, 2013), nicht ausreichend empirisch validiert werden kann (Yee, 2005) und die vier Spielertypen nicht trennscharf voneinander abgegrenzt werden können (Dixon, 2011). Eine Alternative, so Dixon, stellen Play-Personas dar, deren Einsatz jedoch abhängig vom Kontext ist und anwendungsspezifische und wissenschaftlich validiert werden muss (Dixon, 2011, S. 3).

lem Spielertyp ebenfalls existent und nachgewiesen ist (Ferro et al., 2013, S. 1). Die Ergänzung eines solchen Werkzeugs in den GCDP bzw. den GCDR kann – wie bereits erwähnt – sinnvoll sein. Aus der Heterogenität seitens der Persönlichkeiten und damit der Spielertypen von Lernenden folgt, dass die Effekte, die einzelne Game Design-Elemente bei Personen auslösen, individuell verschieden sind. So fühlen sich Socialiser, welche die Findung und den Erhalt von sozialen Kontakten anstreben weniger durch ein Leaderboard herausgefordert bzw. motiviert, als ein Killer, welcher nach Bartle (1996) Wettbewerb mit anderen Spielern anstrebt und sich durch sie herausgefordert fühlt. Eine Zuordnung von Game Design-Elementen zu Spielertypen ist zwar in Tendenzen zu erkennen, jedoch aktuell noch nicht ausreichend wissenschaftlich validiert (Dichev u. Dicheva, 2017) und birgt bei *falschem* Einsatz die Gefahr, die intrinsische Motivation von Lernenden negativ zu beeinflussen (Rughinis, 2013a).

Obwohl bei diesem Arrangement eine derart detaillierte Betrachtung im Vorfeld nicht strukturiert durch einen Fragebogen erfolgte, wurden sehr wohl Vorüberlegungen angestellt, welche Game Design-Elementen bei den Lernenden adäquat und damit vielversprechend für das Gelingen der Intervention waren. Damit einhergehend war es vonnöten, die Lernenden in Bezug auf deren Offenheit für diese Art zu Lernen einzuschätzen, da vergleichbare Experimente in der Vergangenheit zeigten, dass eine zu geringe studentische Offenheit ein weiterer kritischer Faktor ist (Berkling u. Thomas, 2013). Selbige Anforderung und Heterogenität gilt ebenso für das Lehrpersonal, welches bereit sein muss, entsprechend Aufwand und kreative Ideen in das Design eines solchen Arrangements zu investieren (Stott u. Neustaedter, 2013, S. 7).

Obwohl das subjektive Empfinden des Dozenten gepaart mit den studentischen Aussagen als durchweg positiv empfunden wurde, so stellt sich statistisch gesehen die Frage nach der methodischen Strenge des Validierungsinstruments. Durch die qualitative Erhebung in Form eines Feedback-Fragebogens mit einer Stichprobengröße von  $n=13$  Personen kann eine Indikation für den Erfolg des Arrangements erzeugt werden. Diese Indikation sollte jedoch von weiteren empirischen Untersuchungen bestätigt werden. Beispielsweise könnte eine größer angelegte quantitative Studie die gewonnenen Thesen zusätzlich validieren.

Nichtsdestotrotz war das Ziel dieses Arrangements die praktische Untermauerung (inkl. der notwendigen Werkzeuge, welche im Zuge der Theorie erstellt wurden) der theoretischen Vereinbarkeit von kompetenzorientierter Lehre und Spielifizierung in einem hochschulischen Kontext. Zusätzlich wurde die Anwendung der entwickelten Instrumente (GCDP und der GCDR) getes-

tet. Diese Vorhaben konnten damit voll umfänglich umgesetzt werden. Dies schafft damit eine weitere und fruchtbare Basis für zukünftige Forschungsarbeiten.

## 2.6 ZWISCHENFAZIT

Das Game Design umfasst eine Vielzahl an Ansätzen, die – basierend auf der jeweiligen Definition eines Games – unterschiedlich ausgeprägt sein können (Bjork u. Holopainen, 2005, S. 7) und damit auch das Design von Gamification maßgeblich beeinflussen. Alle Ansätze verfolgen jedoch das Ziel „[...] möglichen Spielern eine einzigartige und langanhaltende Spielerfahrung zu bieten“ (Rehfeld, 2014, S. 23) und ein Game hinsichtlich seiner Game (Design) Elemente<sup>32</sup> zu konzipieren oder zu beschreiben. Der dabei wohl wichtigste Aspekt ist nach Dichev et al. der eigentliche Design Prozess für das Design von Gamification, beispielsweise bei einem Einsatz in der Lehre (Dichev et al., 2014, S. 95). Bei einem derartigen Design sollen Lehrende dazu gebracht werden, „[...] to think like a game designer and to consider it as a way of reflecting, as an experience“ (Dichev et al., 2014, S. 95). Ebenfalls erachten Dichev et al. sowie Zichermann u. Cunningham es als wichtig, dass wie bei Games die motivationalen Faktoren ausbalanciert sind und weder extrinsische noch intrinsische Faktoren überwiegen (Dichev et al., 2014, S. 95). Um diese Balance zu erreichen soll sich adäquater und etablierter Game (Design) Elemente bedient werden (Deterding et al., 2013), die basierend auf motivationspsychologischen Theorien kontextabhängig eingesetzt werden und es zulassen, eine Aussage über diesen Wirkungszusammenhang auf Teilnehmende zu treffen.

Dieses Kapitel zeigte dafür wichtige Ansätze und Modelle, die als Grundlage für die folgenden Kapitel dienen. Im Folgenden soll sich einem anderen wichtigen und gleichzeitig grundlegenden Thema neben den strukturellen oder strukturierenden Fragestellungen des Gamification-Ansatzes gewidmet werden: der Motivation. Sie bildet den Kern des Ansatzes und fließt anhand verschiedener Motivationstheorien in den Ansatz ein, welche beim Design derartiger spielifizierter Anwendungen unbedingte und essentielle Beachtung finden müssen (Rigby, 2014, S. 115).

<sup>32</sup> In der Literatur wird ebenfalls der Begriff *Gamebits* synonymisch verwendet (siehe z. B. Rehfeld (2014)).

---

MOTIVATIONSFÖRDERUNG DURCH GAMIFICATION

---

*„There is nothing as practical as a good theory.“*

— Kurt Lewin (Lewin, 1951, S. 169)

### 3.1 EINLEITUNG

Die Potentiale von Spielifizierung werden nicht nur in der einschlägigen Fachwelt wahrgenommen. So zeigt eine Studie der Federation of American Scientists, dass viele Eigenschaften<sup>1</sup> von Games auch in Lernumgebungen angewendet werden können und vice versa Game Designer implizit beim Design von Games Axiome von Lernforschern integrieren (Federation of American Scientists, 1710, S. 4f.). Auch in der nationalen Politik sind diese Wechselwirkungen wahrnehmbar. So enthält der Koalitionsvertrag der 18. Legislaturperiode den folgenden Passus:

„Digitale Spiele prägen den Alltag vieler, insbesondere jüngerer Menschen in unserem Land. Wir erkennen die Vielfalt hochwertiger Angebote, insbesondere pädagogisch wertvoller Computerspiele, sowie die große kreative Leistung und hohe technische Kompetenz der Spieleentwickler an. Dies wollen wir weiter fördern“ (CDU Deutschland et al., 2013, S. 96).

Die Anerkennung, welche den „pädagogisch wertvollen Computerspielen“ durch die Parteien entgegengebracht wird, zielt vor allem auf die Nutzbarkeit von ludischen<sup>2</sup> Games in Lehr-Lernkontexten ab. Derartige Spieler werden ganzheitlich gefördert und eine Aktion sofort mit einem Feedback versehen, was dazu führen kann, dass bis vor Spielbeginn passive Rezipienten während

---

<sup>1</sup> Beispielsweise sind von der Federation of American Scientists angeführt: Motivation und wohldefinierte Zielorientierung, Bereitstellung von Tipps, Hinweisen oder Teillösungen, um das Lernen aufrecht zu erhalten oder zu Trainieren, Personalisierung des Lernens durch Adaption des Mastery Levels, etc. (Federation of American Scientists, 1710, S. 5).

<sup>2</sup> Im Sinne von Caillois und angelehnt an seine Bezeichnung *ludus* (Caillois, 2001, S. 13).

des Spielens zu aktiven und selbstverantwortlich handlungsbereiten Charakteren entwickelt werden (Müller, 2014, S. 69).

Zudem erlaubt der Einsatz derartiger Spielifizierung Lernenden einen geschützten Raum zu bieten, in dem sie Möglichkeiten kreativ erproben bzw. nutzen können, um ihnen gestellte Probleme zu lösen (Grünberger, 2014, S. 6). Erkennbar ist während des Problemlösens eine Verschiebung des Fokus hin zu den Aktivitäten innerhalb des damit einhergehenden Lernprozesses und nicht allein auf dessen Ergebnis (Deterding, 2012, S. 14). Damit nimmt Gamification in Lehr-Lernkontexten nicht nur die Funktion eines Belohnungs- oder Feedbacksystems ein, sondern besitzt einen holistischeren Charakter (Deterding, 2014, S. 306), wie auch Dichev et al. feststellen:

„The gamification of learning is more than reward or a feedback system. It is an approach which is evolving, in parallel with technological developments, to include much larger scales for gameplay, new tools and new ways to connect people“ (Dichev et al., 2014, S. 91).

Das übergeordnete Ziel einer Verhaltensänderung, bewirkt durch eine Motivationssteigerung, stellt einen der Kernpunkte bei einem Einsatz von Gamification oder ludischen Games in lernpädagogischen Kontexten dar. Dieser Sicht auf Gamification soll im Folgenden weiter Beachtung geschenkt werden. Motivation ist für Gamification mehr als eine Triebkraft. Vielmehr ist dieses psychologische Konstrukt bildungssprachlich gesprochen der Kraftstoff, welcher vermag den Motor Mensch anzutreiben oder – bei nicht Vorhandensein – für Stillstand zu sorgen.

Bevor sich im Detail den motivationalen Ausprägungen von Gamification in hochschulischen Lehr-Lernaktivitäten gewidmet werden kann, soll vorab ein grundlegendes Verständnis über die wichtigsten allgemeinen motivationstheoretischen Konzepte gebildet werden. Dies trägt unter anderem dazu bei, gute wissenschaftliche Forschungspraxis im Sinne des DSR nach Hevner et al. (2004) fortzuführen und zunächst die Wissensbasis zu erläutern, um sich anschließend den entsprechenden Anwendungen widmen zu können.

Dieses Kapitel gibt deshalb zunächst einen Überblick über motivationstheoretische Grundlagen und präsentiert Strategien sowie Prinzipien zur Förderung von Lernmotivation auf Basis von spielifizierten Anwendungen in der Lehre. Abschließend werden Forschungsdesiderate abgeleitet.



### 3.2 MOTIVATIONSTHEORETISCHE GRUNDLAGEN

Das Ziel von Gamification ist, nach der Definition dieser Arbeit, Lernende zum Lernen zu motivieren, also die Förderung deren Lernmotivation. Doch was bedeutet es, *motiviert* zum Lernen zu sein? Wie lässt sich dieser Begriff *Motivation* fassbarer machen? Die Motivationsforschung liefert hierzu mannigfaltige und sich in den letzten Dekaden stark verändernde Ansätze und Theorien. Die Wichtigsten und zugleich Beständigsten sollen im Folgenden fokussiert werden.

#### 3.2.1 Begriffsannäherung

Smolka und auch Mazur verstehen unter dem Begriff der Motivation die grundlegende Voraussetzung für *operantes Verhalten*, also für *gute* Performanz und erfolgreiches Lernen (Smolka, 2004, S. IX; Mazur, 2006, S. 231). Smolka postuliert weiterhin, dass Motivation das Verlangen eines jeden Menschen ist, etwas neues zu erschaffen, zu experimentieren oder etwas zu bewirken (Smolka, 2004, S. IX). Dabei kann die Motivation als eine Persönlichkeitseigenschaft oder für jeden Menschen individuelle Disposition aufgefasst werden, die zeitlich begrenzt sein kann und damit einen psychologischen Zustand beschreibt („milde Form der Besessenheit“ (Rheinberg u. Krug, 2005, S. 15, zitiert nach De Charms, 1979, S. 55)) (Woolfolk, 2014, S. 387). Woolfolk ergänzt, dass „[...] jede motivationale Lage ein Produkt aus Eigenschaften und Zustand [ist]“ (Woolfolk, 2014, S. 387). Weiterhin wird die Motivation als ein „richtungsgebender [und] richtungshaltender Prozess“ beschrieben (Mietzel, 2007, S. 344). Dies bedeutet, dass der Motivation nicht alleine eine aktivierende Rolle zugesprochen wird, sondern vielmehr dazu in der Lage ist, diese anfängliche Aktivierung zu steuern und aufrecht zu erhalten. Mit dieser Argumentation folgt Mietzel dem Prozessverständnis, welches dem Naturell der Motivation zu Grunde liegt, was auch von Smolka und Mazur so interpretiert wird.

#### 3.2.2 Zum Motivationsbegriff

Dresel u. Lämmle argumentieren in ähnlicher Art und Weise, detaillieren jedoch die motivationale Lage eines Menschen und unterstellen, dass motivationale Prozesse während des gesamten Handlungsprozesses<sup>3</sup> (Initiierung bis

<sup>3</sup> In diesem Zusammenhang sei das *Rubikon-Modell* von Heckhausen u. Gollwitzer (1987) erwähnt, welches einen menschlichen Handlungsprozess in vier chronologische Prozessphasen unterteilt: *Prädezyonale Phase*, *Präaktionale Phase*, *Aktionale Phase*, *Postaktionale*

Evaluation) zum Tragen kommen. Sie interpretieren das hypothetische Konstrukt der Motivation als eine Kombination aus Zielen und Motiven (Dresel u. Lämmle, 2011). Ziele<sup>4</sup> sind die angestrebte Verkleinerung<sup>5</sup> eines gegenwärtigen *IST*-Zustandes hin zu einem *SOLL*-Zustand (Ergebnis), welche bewusst von Individuen wahrgenommen wird (Locke u. Latham, 1990; Locke u. Latham, 2006). Motive hingegen, beschreiben den Grund bzw. die Gründe für menschliche Aktionen, die zur Zielerreichung beitragen, wobei eine Zielerreichung durch mindestens ein Motiv, dem Menschen bewusst oder unbewusst, angestoßen wird (Dresel u. Lämmle, 2011). Das Wirken der Kombination aus Zielen und Motiven, so Dresel u. Lämmle, sei nicht direkt von außen beobachtbar, sondern kann nur über Indikatoren (angelehnt an Verhalten, Gedanken oder Gefühle) sichtbar gemacht werden (Figas et al., 2013, S. 1300; Dresel u. Lämmle, 2011; Pekrun, 2011). Eine weitere Determinante, welche einen Einfluss auf die Stärke und Qualität von menschlicher Motivation haben kann, sind die individuell unterschiedlichen und damit persönlichen Eigenschaften von Menschen. Sogenannte *personale Bedingungsfaktoren* sind nicht in jeder Situation gleich (instabile Persönlichkeitseigenschaften) und hängen damit von situativen bzw. kontextspezifischen Umwelteinflüssen ab, die in Wechselwirkung zueinander eine Motivation ausbilden. Auch Interessen (also besondere Beziehungen von Menschen zu spezifischen oder abstrakten Objekten oder Tätigkeitsklassen, die während des Erlebens zu positiven Emotionen führen (Dresel u. Lämmle, 2011, S. 105)), die zu den *stabilen Persönlichkeitseigenschaften* von Dresel u. Lämmle gezählt werden,

„[...] sind theoretisch von der aktuellen Motivation für eine spezifische Handlung getrennt und als persönliche Bedingungsfaktoren aufgefasst, die erst in Wechselwirkung mit den Merkmalen einer Situation zur Herausbildung der aktuellen Motivation für eine bestimmte Handlung führen“ (Dresel u. Lämmle, 2011, S. 82).

Dies bedeutet konkret, dass auch Menschen, die sich beispielsweise für das Lösen von Differentialrechnungen grundsätzlich interessieren, unter bestimmten Umgebungseinflüssen nicht dazu motiviert sind, dies zu tun. Es lässt

---

*Phase.* Ist das Ergebnis der ersten Phase (Abwägung bzgl. einer Handlung) ein positives, so wird metaphorisch der Rubikon (Fluss überschritten von Cäsar in dem römischen Bürgerkrieg zwischen Cäsar und Pompeius) überschritten und konkrete Pläne darüber entworfen, wie die Aktion(en) ausgeführt wird.

<sup>4</sup> In der Literatur wird in unterschiedliche *Zieltypen* (wie z. B. *Leistungsziele* Woolfolk (2014, S. 397) oder *Vermeidungsziele* Daumiller et al. (2015)) unterschieden. Für eine ausführliche Diskussion sei auf Werke von Woolfolk (2014) oder Dresel (2010) bzw. Dresel u. Lämmle (2011) verwiesen.

<sup>5</sup> Diese aktionale Verkleinerung zwischen IST- und SOLL-Zustand lässt sich auch in der lateinischen Begriffsabstammung von Motivation erkennen, nämlich des Verbs *movere*, was so viel bedeutet wie bewegen bzw. in Bewegung setzen (Rudolph, 2013, S. 14).

sich folgern, dass sich die Stärke und Qualität einer konkreten Motivation für eine spezifische Handlung ergibt aus einer Wechselwirkung von Zielen und Motiven, die unter Einfluss von Umgebungsbedingungen und individuellen Persönlichkeitseigenschaften bewertet werden (Gage u. Berliner, 1996, S. 338). Die Abbildung 14 veranschaulicht diesen textuell dargestellten Zusammenhang.

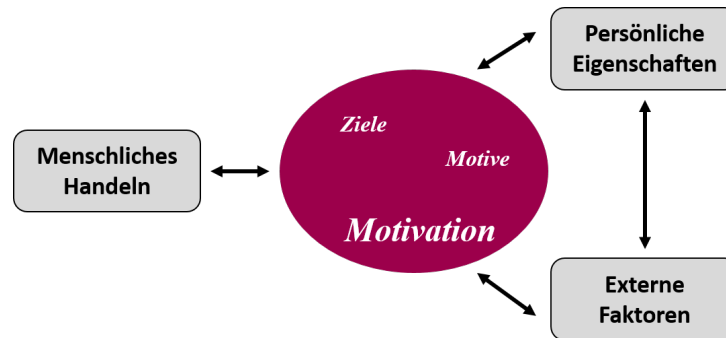


Abbildung 14: Determinanten motivationalen Handelns nach (Figas et al., 2013, S. 1300, eigene Übersetzung) bzw. Gage u. Berliner (1996, S. 338)

Auf Basis dieser Annahmen und in Anlehnung an das *Rubikon-Modell* definieren Dresel u. Lämmle Motivation wie folgt:

„Motivation ist ein psychischer Prozess, der die Initiierung, Steuerung, Aufrechterhaltung und Evaluation zielgerichteten Handelns leistet“ (Dresel u. Lämmle, 2011, S. 81).

Nach näherer Betrachtung der unterschiedlichen Aspekte, welche die Motivation für eine Handlung beeinflussen können, wird klar, dass es sich bei der Motivation um ein höchst individuelles und situationsabhängiges Konstrukt handelt, für dessen Erzeugung bzw. Förderung es kein allgemeines Patentrezept geben kann (Dresel, 2010; Rheinberg u. Krug, 2005; Smolka, 2004). Es existieren durchaus teilweise erprobte und wissenschaftlich validierte Empfehlungen (im Sinne von *best-practices*), die sich als wirksam für die Motivationsförderung von Individuen herausgestellt haben (siehe z. B. Rheinberg u. Krug (2005)). Allerdings sind diese kein Garant dafür, dass sie in einem anderen Kontext, abweichend von dem Erprobungskontext ebenso gut funktionieren und den gleichen Effekt erzielen.

### 3.2.3 *Intrinsische und extrinsische Motivation*

Eine häufig in der Literatur vorzufindende Unterscheidung von Motivationsarten, ist die Unterteilung in *intrinsische* und *extrinsische* Motivation (siehe z. B. Deci u. Ryan 1985; Deci u. Ryan 1993; Pekrun 2011; Smolka 2004; Wool-

folk 2014). Diese Unterscheidung ist ebenso gültig für die Lernmotivation (Waldherr u. Walter, 2009). Gamification kombiniert diese beiden Motivationsarten durch extrinsische Anreize, wie Level oder Punkte und intrinsische Anreize, zu denen das Gefühl Dinge zu meistern („*Mastery*“), Autonomie oder sozialer Zugehörigkeit gehört (z. n. Richter et al., 2015, S. 24).

**INTRINSISCHE MOTIVATION** „[...] ist die natürliche Tendenz, sich Herausforderungen auszusuchen und sie zu meistern, während persönlichen Interessen nachgegangen wird und Fähigkeiten umgesetzt werden“ (Woolfolk, 2014, S. 387). Dabei werden keine Anreize oder Bestrafungen benötigt, da entweder Interesse für ein Thema besteht oder die Tätigkeit als solche positive Emotionen hervorruft, die als Belohnung fungieren (Dresel u. Lämmle, 2011, S. 89) (Woolfolk, 2014, S. 387). Intrinsisch Lernende lernen nachhaltiger und sie nehmen sich als neugierig, wissbegierig, aktiv und interessiert wahr (Deci u. Ryan, 1993, S. 225). Beispielhaft sei hier das bereits angeführte Szenario genannt, bei dem durch persönliches Interesse getrieben, Differentialgleichungen gelöst werden. Während des Lösen und nach Abschluss der Rechnung wird ein Bearbeiter mit positiven Emotionen belohnt, die individuell stark ausgeprägt sein können.

**EXTRINSISCHE MOTIVATION** hingegen fokussiert nicht den Wert einer Handlung selbst, sondern vielmehr deren Ergebnis (Dresel u. Lämmle, 2011). Demnach sind alle Aktivitäten, die auf den Anreiz von Zielen oder Ergebnissen gerichtet sind, extrinsisch (Rheinberg, 2008, S. 149). Bezogen auf das Beispiel der Differentialgleichungen würde dies bedeuten, dass beispielsweise das Motiv zum Lösen einer derartigen Aufgabe fremdbestimmt durch eine Lehrkraft erzeugt bzw. beeinflusst wurde, unter Einbindung einer „Belohnungs- oder Sanktionierungsstruktur“ (Dresel u. Lämmle, 2011, S. 89).

Intrinsische und extrinsische Motivation schließen sich ebenso wenig aus wie sie jeweils als Reinform vorkommen. Eine Handlung kann beispielsweise nicht als ausschließlich intrinsisch motiviert diagnostiziert werden und damit einem *alles-oder-nichts-Prinzip* folgen. Vielmehr kann jede Handlung auf einem Kontinuum von völlig selbstbestimmt (intrinsisch) bis hin zu völlig fremdbestimmt (extrinsisch) verortet werden (Woolfolk, 2014, S. 388). Kombinationen aus beiden Motivationsarten sind damit nicht nur existent, sondern durchaus wünschenswert und spielen vor allem beim Lernen eine entscheidende Rolle.

### 3.2.4 *Einschlägige Motivationstheorien*

Nachdem ein Verständnis über die allgemeinen Begriffe der *Motivation* gebildet wurde, sollen nun für Gamification relevante Motivationstheorien identifiziert und herausgearbeitet werden. Die Einschränkung erfolgt hier anhand der Vorarbeit von Schlagenhauser u. Amberg (2014), welche in einem Literature Review von insgesamt 34 Artikeln von neun Autoren aus 30 darin vorkommenden psychologischen Theorien sechs identifizieren konnten, die nach ihrer Ansicht am Häufigsten dem Konzept von Gamification zu Grunde gelegt werden und damit ein Indiz für deren Wichtigkeit darstellen (Schlagenhauser u. Amberg, 2014). Wichtig ist dabei zu erwähnen, dass jeder Ansatz einer speziellen Motivationstheorie auch eine Sicht auf Motivation darstellt, die nicht überschneidungsfrei zu weiteren Theorien abgegrenzt werden kann. Vielmehr sind inhaltliche Schnittmengen zwischen den Theorien zu erkennen (Schlagenhauser u. Amberg, 2014). Auch impliziert diese Auswahl ebenso wenig, dass nicht noch weitere Theorien existieren oder existierende Theorien derartig anpassen lassen, dass sie den Gamification-Ansatz aus motivationstheoretischer Perspektive zusätzlich unterstützen könnten. Die Tabelle 3 gibt eine Übersicht über die identifizierten Theorien und deren zu Grunde liegenden Sichtweisen sowie wesentlichen Charakteristika.

Theorie	Hauptvertreter	Motivationsquelle	Wichtige Einflüsse
Flow-Theory	Csikszentmihalyi (2010)	intrinsisch	Angemessenheit von Herausforderungen und Fähigkeiten
Self-Determination Theory (Humanistisch)	Deci u. Ryan (1985), Maslow (2013)	intrinsisch	Bedürfnis nach Selbstwert, Selbsterfüllung, Selbstbestimmung
Self-Efficacy Theory (Sozialkognitiv)	Bandura (1977)	intrinsisch & evtl. extrinsisch	Individuelle Selbstwirksamkeitsbeurteilung
weiter auf nächster Seite			

Theorie	Hauptvertreter	Motivations- quelle	Wichtige Ein- flüsse
Theory of Planned Behavior (Sozialpsychologisch)	Ajzen (1991)	intrinsisch	Verhalten, Volition, Intention (inkl. Determinanten)
Uses and Gratification Theory	Katz et al. (1973)	intrinsisch	Persönliche Motive und Interessen

Tabelle 3: Übersicht über für Gamification relevante psychologische Theorien (angelehnt an Schlagenhauer u. Amberg (2014, S. 7) und Woolfolk (2014, S. 390))

#### 3.2.4.1 *Flow Theory nach Csikszentmihalyi*

Der Glücksforscher und Psychologe Csikszentmihalyi entwickelte durch die Beobachtung von Personen aus verschiedensten Lebensbereichen<sup>6</sup> eine Theorie, die den Menschen in einem Trance-ähnlichen Zustand beschreibt (Csikszentmihalyi, 2010). Bei diesen Observationen stellte Csikszentmihalyi fest, dass ein Gefühl existiert, welches Menschen antreibt, wiederholt an die Grenzen ihres Leistungsvermögens zu gehen und sich dabei neuen Herausforderungen zu stellen (Csikszentmihalyi, 2010). Huhn formuliert hierzu:

„Als *Flow* gilt in der Psychologie eine Erfahrung, bei der das Subjekt völlig in einer Tätigkeit aufgeht und dabei ein besonderes Glücksgefühl des Gelingens erlebt. Das Streben nach dieser Erfahrung bewegt Menschen, strapaziöse und mit viel Einsatz verbundene Tätigkeiten um ihrer selbst willen auszuführen“ (Huhn, 2004, S. 248).

Dabei dringen jegliche von Formen von Missempfindungen oder externe Stimuli (z. B. Reize, wie Durst oder Müdigkeit) nur schwer in das Bewusstsein oder werden schlichtweg ignoriert (Csikszentmihalyi, 2010). Auch der Verlust des Zeitgefühls geht in der Regel mit der starken Fokussierung auf eine Aufgabe einher (Csikszentmihalyi, 2010). Das gänzliche Aufgehen in der Tätigkeit sorgt neben einem glatten und fließenden Handlungsverlauf auch für das Ausbleiben der Besorgtheit über ein mögliches Scheitern (Langner u. Mertens, 2012, S. 18). Außerdem wird nach erfolgreichem Abschluss einer Herausforderung eine Persönlichkeitsentwicklung von Huhn nachgesagt, die

<sup>6</sup> Beispielsweise Tänzer, Ärzte oder Bergsteiger.

einher geht mit psychischer Sicherheit bzgl. der inneren Ordnung und Struktur, welche dem Erlernem während der Tätigkeit geschuldet ist (Huhn, 2004).

Die Grundidee besteht nach Csikszentmihalyi also darin, die erfahrbare Lebensfreude zu maximieren, wenn die zu bewältigenden Herausforderungen unter Berücksichtigung der individuellen Fähigkeiten in einem subjektiv-optimalen Verhältnis zueinander stehen (siehe Abbildung 15). Ist diese optimale Angemessenheit erreicht, befindet sich ein Mensch in einem Flow-Kanal, dem Inbegriff einer positiven Sucht (Csikszentmihalyi, 2010). Andernfalls, wenn beispielsweise die Fähigkeiten zu hoch für die gestellten Herausforderungen sind, droht eine Unterforderung, was zum Gefühl der Langeweile führt (Csikszentmihalyi, 2010). Im umgekehrten Fall, also wenn die Herausforderungen zu hoch im Vergleich zu den individuellen Fähigkeiten sind, erzeugt dies eine Überforderung, was sich meist im Gefühl von Angst, Stress oder gar Enttäuschung äußert (Csikszentmihalyi, 2010; Huhn, 2004).

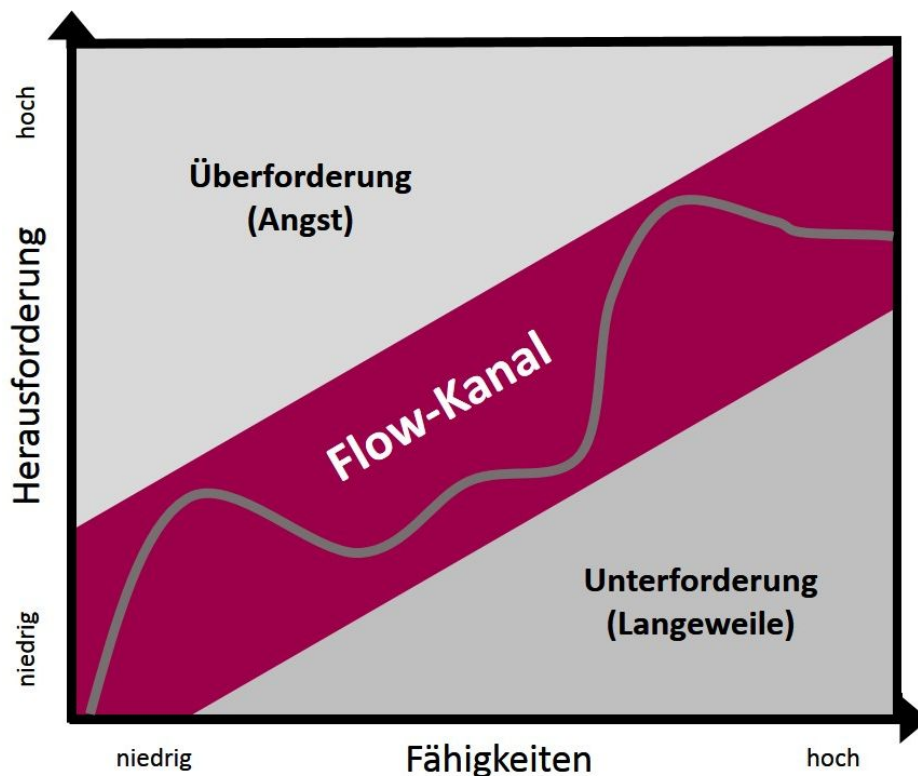


Abbildung 15: Flow Theory nach Csikszentmihalyi (2010, S. 74, eigene Darstellung)

Um im Flow-Kanal zu wachsen, also ein höheres Level an Fähigkeiten oder Herausforderung erfolgreich absolvieren zu können, kommt es darauf an, welche der beiden Komponenten zuerst erhöht werden soll. Werden also neue Fähigkeiten erlernt, so muss damit auch die Herausforderung steigen (und vice versa), um *im Flow zu bleiben* (Rehfeld, 2014, S. 174). Für den Flow-Kanal gilt

ebenso, dass es sich um einen Bereich handelt und nicht um ein lineares Optimum. Die in Abbildung 15 angedeutete graue und geschwungene Linie durch den Flow-Kanal deutet diesen Umstand an. Das Eintreten in diesen Kanal ist jedoch keineswegs einfach oder gar mühelos. Es bedarf nach Csikszentmihalyi häufig „[...] schwerer körperlicher Anstrengung oder einer hoch disziplinierten geistigen Aktivität“ (Csikszentmihalyi, 2010, S. 81). Weiterhin müssen nach Huhn die folgenden externen und persönlichkeitsbezogenen Voraussetzungen für das Erleben einer Flow-Erfahrung, die prinzipiell bei jeder Aktivität auftreten kann, erfüllt sein (Huhn, 2004, S. 249ff.):

- **Ziele:** Für das Erfahren von Flow, müssen ein oder mehrere Ziele existieren, die von der jeweiligen Person im Vorfeld selbst definiert wurden. Ziele sollten dabei operationalisierbar, messbar und ebenso erreichbar sein (Dresel u. Lämmle, 2011).
- **Feedback:** Eine wichtige Voraussetzung für das Eintreten in den Flow-Kanal ist das kontinuierliche und unmittelbar nach einer Aktion ausgeführte jeweilige Feedback. Es gibt Aufschluss darüber, ob man erfolgreich bei der Verwirklichung eines Ziels ist. Dabei ist jeder einzelne Schritt ebenso entscheidend, wie die Zielerreichung selbst.
- **Angemessenheit der Schwierigkeit:** Wie bereits erwähnt, ist das Eintreten in den Flow-Kanal nur dann möglich, wenn ein subjektiv empfundenes optimales Verhältnis zwischen gestellten Herausforderungen und vorherrschenden Fähigkeiten besteht.
- **Autonomie und Steuerung:** Der betreffenden Person wird das Gefühl nach Steuerung und Kontrolle hinsichtlich der Zielerreichung gegeben - eine Voraussetzung, die auch in der SDT nach Deci u. Ryan so besteht (Deci u. Ryan, 1985; Deci u. Ryan, 1993).
- **Konzentration:** Eine hohe Konzentration und Fokussierung auf die ausführende Aktion ist erforderlich, um die Stimuli zu begrenzen und damit andere, nicht aktionsabhängige Dinge auszublenden.

Games haben das Potential durch *gutes* Game Design diesen Flow-Zustand bei einem Spieler zu erzeugen (Huhn, 2004; Schell, 2015). Dabei tragen alle Game Design-Elemente gleichermaßen zu der Entstehung von Flow bei, bei dem sich der Spieler in einem Spielfluss befindet und die oben genannten Symptome (z. B. Verlust des Zeitgefühls) auftreten können (Rehfeld, 2014, S. 57ff., S. 72f., ). Auch Gamification kann diesen immersiven Zustand herstellen, in dem beispielsweise einzelne Game Design-Elemente so ausgestaltet sind, dass sie



einen Spieler fordern, aber nicht überfordern und seiner persönlicher Kompetenz angemessen sind (Dichev et al., 2014, S. 89; Rigby, 2014, S. 116). So sollten bei der Erstellung von Gamification-Konzepten oben genannte Voraussetzungen mit in die Ausprägungen der verschiedenen Game Design-Elemente und die Entscheidung bzgl. deren Kombination einfließen (Sillaots, 2014; Hamari u. Koivisto, 2014). Beispielsweise können Feedback-Mechanismen, wie Levels oder Fortschrittsanzeigen, Auskunft darüber geben, welche Schritte hin zu einer Zielerreichung noch notwendig oder generell möglich sind. Das würde implizieren, dass Spieler mehrere Handlungsmöglichkeiten (definiert durch Regeln) innerhalb einer Story bzw. Erzählung oder eines Lernszenarios besitzen, (Lern-)Ziele zu erreichen und diese gemäß ihren eigenen Fähigkeiten wählen können. Huhn weißt jedoch darauf hin, dass das Erleben von Flow nicht alleinig durch die Inaussichtstellung einer Belohnung (extrinsisch) erreicht werden kann (Huhn, 2004, S. 8). Vielmehr bedarf es intrinsischer Motivation für die Sache. Extrinsische Anreize können aber dazu beitragen, dass Glücksgefühle erzeugt werden, die Personen, welche bereits im Flow-Kanal sind, dort verbleiben (Huhn, 2004, S. 8). Zichermann u. Cunningham vertreten sogar die These<sup>7</sup>, dass fortwährend erfahrene Erfolge, motiviert durch extrinsische Anreize, sich in intrinsische Motivation transformieren lassen (Zichermann u. Cunningham, 2011). Dörnyei u. Ushioda unterstützen diese These und attestieren der intrinsischen Motivation eine Ergänzung durch extrinsische Motive, sofern diese ausreichend internalisiert wurde (Dörnyei u. Ushioda, 2013). Dies widerspricht zwar der SDT, die selbiges nur anders herum feststellt, jedoch würde es erklären, warum Menschen, die sich beispielsweise mit dem Verständnis eines Themas Schwierigkeiten haben, sich aber lange genug damit auseinandersetzen und schließlich nach erfolgreicher Verinnerlichung Spaß daran finden und damit den Zustands des Flow erreichen können.

#### 3.2.4.2 *Deci und Ryans Self-Determination Theory*

Die SDT (zu deutsch: Selbstbestimmungstheorie) setzt auf der *Hierarchie der Bedürfnisse* von Maslow (2013) auf und legt zu Grunde (Dichev et al., 2014, S. 82), dass

<sup>7</sup> Neben Zichermann u. Cunningham, welche sich für einen Einfluss von extrinsischen Anreizen auf intrinsische Motivation aussprechen, steht die Argumentation von Mekler et al., welche in einer Studie herausfanden, dass der Einsatz von vornehmlich extrinsisch einzuordnenden Game Design-Elementen (Punkte, Levels und Leaderboards), keinen Effekt auf die intrinsische Motivation von Probanden in der Studie hatten (Mekler et al., 2013b). Der Einfluss des Kontextes mag ein ausschlaggebendes Kriterium für die unterschiedlichen Interpretation von Wechselwirkungen beider Motivationsarten sein.

„[...] alle Menschen sich kompetent und fähig in ihrem Umgang mit anderen Menschen fühlen, über einen gewissen Entscheidungsspielraum und Kontrolle in ihrem Leben verfügen und dabei auch Beziehungen zu anderen Menschen aufnehmen und aufrecht erhalten [...]“ (Woolfolk, 2014, S.392).

Somit unterstellen Deci u. Ryan jedem Menschen angeboren drei psychologische Grundbedürfnisse (*Kompetenz*, *Autonomie* und *soziale Eingebundenheit*), die universell und damit auch kulturübergreifend sind (siehe Abbildung 16) (Deci u. Ryan, 1985; Deci u. Ryan, 1993; Jang et al., 2009).

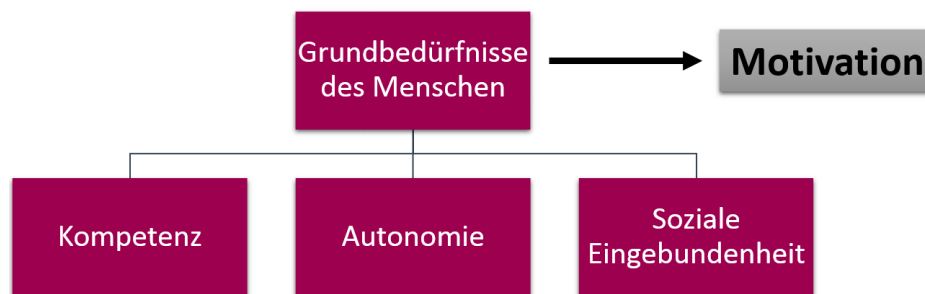


Abbildung 16: SDT nach Deci u. Ryan (1985)

Sie sind nach der Ansicht von Ryan u. Deci essentiell

„[...] for facilitating optimal functioning of the natural propensities for growth and integration, as well as for constructive social development and personal well-being“ (Ryan u. Deci, 2000b, S. 68).

Die drei Grundbedürfnisse sind sowohl für die intrinsische als auch extrinsische Motivation gleichermaßen relevant. Das Bedürfnis der Kompetenz oder auch Wirksamkeit entspricht dem individuell durch den Menschen wahrgenommenen Erfolg beim Wirken der Ausübung spezieller Aktionen oder Verhaltensweisen (Deci u. Ryan, 1985; Deci u. Ryan, 1993). Die Autonomie definiert den dabei wahrgenommenen Grad an Selbstbestimmung (Freiwilligkeit) des eigenen Handelns, während die soziale Eingebundenheit (Bedeutung seiner selbst für Andere und Andere für einen selbst) die individuelle Wahrnehmung selbiger ist (Deci u. Ryan, 1985; Deci u. Ryan, 1993).

Die Intensität und die Art und Weise, wie diese Grundbedürfnisse befriedigt werden können, ist eine individuelle Fragestellung und unter Anderem abhängig von dem jeweiligen Kontext, in dem sich eine Person befindet (Ryan u. Deci, 2000b). Im Zuge ihrer Forschungen, fanden Deci u. Ryan heraus, dass ein Kontext, der es zulässt, das Autonomiebedürfnis eines Menschen, bei der Erfüllung von individuellen Zielen (intrinsisch motiviert), zu befriedigen, nicht von extrinsischen Anreizen abhängig ist, ja diese sogar eine vorhandene intrinsische Motivation regelrecht untergraben können (Ryan u. Deci, 2000b,

S. 70). Damit wird das Autonomiebedürfnis in den Mittelpunkt der Selbstbestimmung gerückt. Dieses Grundbedürfnis, in Verbindung mit einer strukturierten Umgebung, vermag nach Jang et al. (2010) eines der mächtigsten Antriebe für eine positive Entwicklung von Lernenden hinsichtlich Engagement, Zufriedenheit oder Kompetenz, angefangen von der Primar- bis zur Tertiärbildung, zu sein (Jang et al., 2010, S. 589ff.).

Die in der Selbstbestimmungstheorie aufgestellten Grundbedürfnisse lassen sich auf Gamification übertragen, bzw. in dem Konzept wiederfinden. Das Grundbedürfnis der Kompetenz wird dadurch genüge getan, den Spieler in die Lage zu versetzen, effektiv auf für ihn neue Spielsituationen reagieren zu können. Eine wesentliche Voraussetzung in einem Lernkontext ist hierbei nicht nur die Tatsache, dass die Herausforderungen entsprechend der Fähigkeiten bzw. Kompetenz eines Lernenden gewählt sind, sondern auch, dass diese Herausforderungen ansprechende und interessante Ziele (im besten Fall Lernziele) repräsentieren, die für den Lernenden mit einem angemessenen Feedback<sup>8</sup> bzgl. der Erreichung dieser Ziele einher gehen (Dresel u. Lämmle, 2011; Groh, 2012). Ist dies der Fall, so spricht Koster von dem Aufkommen von *Mastery*, und meint das stetige (teilweise subtile und unbewusste) Verbessern der eigenen Fähigkeit in Verbindung mit positiven Emotionen, wie Spaß o.Ä. (Koster, 2013).

Die Freiheit, zu entscheiden, welche Spielaktion ausgeführt werden soll und wie die Ausführung aussieht, ist ein wichtiges Kriterium sowohl in Games als auch im Konzept von Gamification. Jedoch treffen hier zwei kontextabhängige Thesen aufeinander. Groh sieht eine Gefahr darin, dass die Autonomie verloren geht, sobald das Gefühl von Kontrolle für einen Spieler aufkommt Groh (2012). Diese These scheint zunächst einleuchtend, jedoch sieht er den Kontrollverlust alleinig dadurch gegeben, dass einem Spieler die Konsequenz einer Handlung vor Augen geführt wird, als Form eines extrinsischen Anreizes (Groh, 2012, S. 42). Dichev et al. sehen genau darin einen Vorteil des Gamification-Ansatzes. Sie geben an, dass Spieler zu jeder Aktion ein formatives Feedback erhalten

[...] whether they have succeeded or not, and how they might improve. At the same time, with experience points, levels, progress bars, a log of completed 'quests' and so on, the players always know exactly where they stand, and what they still have to do to get to their goals“ (Dichev et al., 2014, S. 89).

<sup>8</sup> Groh spricht von *juicy* Feedback, also Feedbackmechanismen, die Interaktionen zwingend an eine Form von teilweise visueller Rückkopplung an den Nutzer binden, um den Erfolg oder Fortschritt widerzuspiegeln und idealerweise für weitere Aktionen anzuregen (Groh, 2012, S. 42f.).

Es gilt also die Autonomie eines Spielers nur insofern zu beschränken, als dass der Spieler motiviert ist, vorgegebene Ziele zu erreichen.

Aber ist die Vorgabe von Zielen nicht alleinig schon eine Einschränkung der Autonomie? – Dies mag durchaus debattierbar sein. Dieser Gratweg zwischen Kontrolle und Autonomie lässt schlussfolgern, dass eine einseitige Betrachtung nicht zielführend ist. Vielmehr gilt es die richtige Balance zwischen so viel Autonomie wie möglich, aber so wenig Kontrolle wie nötig zu finden. Diese Dualität und das Wirken auf die Autonomie und sogar auf die Kompetenz wurde in der *kognitiven Bewertungstheorie* von Deci u. Ryan näher erläutert (Deci u. Ryan, 2002). Sie schlagen vor, dass jedes Ereignis mit zwei Aspekten verbunden ist, dem *Kontroll-* und *Informationsaspekt*. Der *Kontrollaspekt* folgt der Argumentation von Groh und wirkt kontrollierend (freiheits-einschränkend) und damit intrinsisch demotivierend (Deci u. Ryan, 2002). Der *Informationsaspekt* hingegen kann das Kompetenzzempfinden steigern und damit die These von Dichev et al. unterstützen (Deci u. Ryan, 2002). Beispielhaft und zur Verdeutlichung sollen zwei Phrasen aus dem Schulalltag von Woolfolk dienen (siehe Tabelle 4).

<b>Kontrollaspekt</b>	<b>Informationsaspekt</b>
„Das ist gut; siehst Du, wenn du meinen Anweisungen folgst, kriegst du eine Eins!“	„Das ist gut; du verstehst jetzt viel besser, wie der Autor die Metaphern verwendet. Du hast die Eins verdient!“

Tabelle 4: Beispiele für die Gegensätzlichkeit von Kontroll- und Informationsaspekten in der kognitiven Bewertungstheorie (Woolfolk, 2014, S. 393)

Im Game Design von Games und ebenso Gamification können Möglichkeiten geschaffen werden, das Gefühl von sozialer Eingebundenheit für den Spieler zu erzeugen. Interessensgemeinschaften, also Gruppen von Menschen, die dieselben Interessen bzw. dieselbe Begeisterung für eine Sache teilen, sind nach Groh hierfür geeignet (Groh, 2012). Sie verbinden einzelne Personen durch die Basis eines gemeinsamen Ziels und Möglichkeiten, während der Zielerreichung zu kommunizieren oder die Wahrnehmung eines Einzelnen innerhalb einer Gruppe zu verändern (Waldherr u. Walter, 2009, S. 90). Letzteres ist beispielsweise durch den Einsatz von Game Design-Elementen möglich, die Status oder Reputation zeigen, also beispielsweise Badges oder Levels. Die Schwierigkeit dieses Bedürfnis im Spiel – egal ob online oder offline – zufrieden zu stellen, liegt dabei in der Findung einer gemeinsamen Basis, bzw. eines gemeinsamen Ziels, das *meaningful* (bedeutsam) für die Spieler ist

(Mandl u. Hense, 2014; Nicholson, 2015; Salen u. Zimmerman, 2004). Bei einem Ziel muss es sich nicht unbedingt um ein Leistungsziel handeln. Ebenso wäre die Bearbeitung einer Lernaufgabe in Gruppenarbeit ein mögliches Ziel, bei dem die Gruppenmitglieder als Teil einer Story ein Quest bearbeiten und dabei ihren Kompetenzen entsprechend unterschiedliche Rollen einnehmen können, die wiederum für sich relevante (selbst-gesteckte) Teilziele beinhalten und jeweilig in der sozialen Gemeinschaft einer Gruppe honoriert werden (Mandl u. Hense, 2014; Rigby u. Ryan, 2011). Zusätzlich kann eine emotionale Bindung zwischen Lernenden und Lehrenden dieses Gefühl verstärken und ebenso das Kompetenzzempfinden bei Lernenden steigern (Waldherr u. Walter, 2009, S. 90).

Abschließend lässt sich festhalten, dass Gamification durchaus den Anforderungen gerecht wird, die durch die SDT gestellt werden. Sicherlich muss hierfür ein entsprechendes Game Design vorliegen. Was *entsprechend* in diesem Fall bedeutet, also wie die Balance zwischen intrinsischer und extrinsischer Motivation in einem Game Design aussehen sollte, darüber sind sich führende Forscher uneins, und existierende Thesen hierzu sind bisher noch nicht ausreichend empirisch validiert (Seaborn u. Fels, 2015, S. 8f.). Dennoch können Theorien wie die SDT wertvolle Hinweise<sup>9</sup> liefern und mögliche, wenn auch marginal scheinende Details aufdecken, deren Summe zu positiven Emotionen und damit Motivation führt, welche für erfolgreiches und nachhaltiges Lernen unabdingbar sind, so Dichev et al.:

„As the games are directly related to human psychological needs and behavioral patterns, they become highly powerful tools for achieving goals in non-entertainment contexts“ (Dichev et al., 2014, S. 89).

### 3.2.4.3 Banduras Selbstwirksamkeitstheorie

Bei dem psychologischen Konstrukt der Selbstwirksamkeit beschreibt Bandura die subjektive Überzeugung einer Person, mit den eigenen Kompetenzen bestimmte Handlungen in herausfordernden Situationen erfolgreich ausführen zu können (Bandura, 1977; Bandura, 1997). Dabei wird der Person ein Gefühl von persönlicher Wirksamkeit vermittelt, welche dazu führen kann, selbstwertdienliche oder motivationsförderliche Attributionen zu erzeugen und diese trotz Hindernissen aufrecht zu erhalten (Bandura, 1977, S. 191; Ban-

<sup>9</sup> Eine Übersicht über Möglichkeiten die Selbstbestimmung und Autonomie und damit die intrinsische Motivation zu erhöhen, ist beispielsweise in Werken von Woolfolk (2014, S. 394f.), Malone u. Lepper (1987, S. 224ff.), Raffini (1996) oder Reeve (1996) zu finden. Diese Empfehlungen sind allgemein gehalten, lassen sich aber damit auf den Bereich der Hochschule adaptieren.

dura, 1997, S. 3; Schwarzer u. Jerusalem, 2002, S. 30). Bandura formuliert hierzu:

„Perceived self-efficacy is concerned not with the number of skills you have, but with what you believe you can do with what you have under a variety of circumstances“ (Bandura, 1997, S. 37).

Schwarzer u. Jerusalem bezeichnen dieses theoretische Konstrukt der Selbstwirksamkeitserwartung als einen „Spezialfall von Optimismus“ (Schwarzer u. Jerusalem, 2002, S. 29). Diese Form von Optimismus bezieht sich allerdings auf Aufgaben oder Aktivitäten, die nicht alleine durch Routine lösbar sind, sondern Anstrengung und Ausdauer erfordern (Schwarzer u. Jerusalem, 2002, S. 35). Bandura (1977) legt zur Bestimmung der *Qualität* von persönlicher Selbstwirksamkeit die Dimensionen *Ausmaß* („magnitude“ (Bandura, 1977, S. 194)), *Stärke* („strength“ (Bandura, 1977, S. 194)) und *Allgemeingültigkeit* („generality“ (Bandura, 1977, S. 194)) an, welche mit aufgabenspezifischen Fragebogennitems erhoben werden können (Bandura, 1977, S. 194; S. 205):

Das *Ausmaß* der Selbstwirksamkeit ist abhängig von der Schwierigkeit einer bestimmten Aufgabe und wird demnach von einzelnen Individuen unterschiedlich beurteilt. Selbstwirksamkeitserwartungen variieren ebenso in puncto *Stärke*. Diese gibt an wie hoch die empfundene Sicherheit einer Person, bezogen auf die Ausführung einer Aufgabe ist. In der Dimension *Allgemeingültigkeit* sieht Bandura die Transferierbarkeit der Selbstwirksamkeitserwartung auf Aktivitäten, die über den Bezugskontext einer einzelnen Aufgabe hinaus gehen. Dabei ist erwähnenswert, dass die Beurteilung der Selbstwirksamkeitserwartung im Vorfeld<sup>10</sup> der Ausführung einer Aufgabe oder Aktivität erfolgt (Zimmerman, 2000, S. 84) und sie durch mehrmaliges erfolgreiches Durchlaufen von Aufgabenbearbeitungen iterativ gesteigert werden kann (Gist u. Mitchell, 1992, S. 190). Die beim mehrmaligen und erfolgreichen Durchlaufen gemachten Erfahrungen (Mastery) haben den stärksten Einfluss auf die Beurteilung des persönlichen Kompetenzzempfindens, sofern sie auf die persönliche Anstrengung hin zurückgeführt werden können (Schunk, 1995). Dieses Empfinden spielt demnach für die akademische Motivation und die damit verbundenen Lern- bzw. Leistungsprozesse eine grundlegende Rolle (Schwarzer u. Jerusalem, 2002, S. 30), denn „[e]ffective functioning requires both skills and the efficacy beliefs to use them“ (Bandura, 1997, S. 36).

<sup>10</sup> Bandura unterscheidet in diesem Zusammenhang in *Konsequenzerwartungen* (*outcome expectancies*) (Bandura, 1977, S. 193), also solche Erwartungen, die „[...] sich auf das für bestimmte Resultate notwendige Verhalten“ (Schwarzer u. Jerusalem, 2002, S. 35) beziehen und *Kompetenzüberzeugungen* (*perceived self-efficacy*) (Bandura, 1977, S. 193), was unter Selbstbezug, die „[...] Frage nach der persönlichen Verfügbarkeit von Handlungen“ (Schwarzer u. Jerusalem, 2002, S. 36) stellt.

Bezogen auf Gamification konnten beispielhaft Banfield u. Wilkerson zeigen, dass eine auf Gamification basierte Lehrform in der Lage ist, die empfundene Selbstwirksamkeit von Lernenden zu steigern (Banfield u. Wilkerson, 2014, S. 297). Im Kontext von System Engineering bzw. Information Assurance kam jeweils eine spielifizierte Lehrform zur Anwendung, die auf der *Experimental Learning Theory* nach Kolb (2015) beruhte. Nach Durchführung der entsprechenden Lehr-Lerneinheiten und anschließender qualitativer Interviewstudie mit den annähernd einhundert Lernenden konnten die Forscher einen starken Zuwachs an intrinsischer Motivation und empfundener Selbstwirksamkeitserwartung feststellen (Banfield u. Wilkerson, 2014, S. 295). Dabei war laut Banfield u. Wilkerson die extrinsische Motivation der Lernenden wesentlich geringer als bei einer nicht spielifizierten Lehrform (Banfield u. Wilkerson, 2014, S. 295). Diese Feststellung ist insofern bemerkenswert, als dass die bloße Existenz von Spielifizierung schon eine Steigerung der intrinsischen Motivation bei den Lernenden hervorgerufen haben muss, wobei diese Motivationsart aus Sicht der Verhaltensanalyse nicht beobachtbar ist (Conor et al., 2014, S. 100). Die Frage, welchen Einfluss<sup>11</sup> das Thema an sich auf die Motivationsarten von Lernenden gehabt hat, bleibt jedoch unbeantwortet. Darüber hinaus bleibt unerwähnt, wie hoch die Identifikation der Lernenden mit durch Lehrende festgesetzte Lernziele in Bezug auf deren persönliche Zielvorstellungen war (z. B. Lernnahziele, die innerhalb einer Lerneinheit als persönlicher Zielmaßstab gelten). Es bleibt jedoch die These im Raum, welche eine Analogie zwischen zwei unterschiedlichen Domänen auf Basis der Selbstwirksamkeitstheorie herstellt: Ein Spieler wird ein Spiel oder eine Aufgabe innerhalb eines Spiels nur dann beginnen, wenn er sich in der Lage fühlt, diese erfolgreich zu meistern (Pavlas et al., 2010, S. 2399). Analog kann dieser Satz für Lernende in Bezug einer akademischen Aufgabenstellung formuliert werden. Pavlas et al. (2010) attestieren der auf Games bezogenen Selbstwirksamkeitstheorie sogar, dass sie in einem direkten Zusammenhang mit dem Erreichen des Flow-Zustandes steht (Pavlas et al., 2010, S. 2401):

„Videogame self-efficacy predicts flow, which predicts motivation; consequently, videogame self-efficacy is actually a key component for ensuring motivation in games because of its significant relationship to flow state“.

<sup>11</sup> Diese fehlende Ausdifferenzierung der verschiedenen Einflussfaktoren ist häufiger in der Literatur vorzufinden und stößt bei Forschern auf Kritik (Landers et al., 2015, S. 6): „Instead of considering the specific attributes of gamification that led to this success, [...] the relationship between the use of the intervention as a whole and outcomes of interest [is examined]“.

#### 3.2.4.4 Theory of Planned Behavior nach Ajzen

Die *Theorie des geplanten Verhaltens* (engl. Theory of Planned Behavior) basiert auf der Grundannahme, dass das Verhalten einer Person gegenüber einer Aufgabe, anderen Person oder einem Sachverhalt vorhersagbar ist (Ajzen, 1991). Sie gehört nach Schlagenhauser u. Amberg zu den wichtigsten Frameworks, um das menschliche Sozialverhalten zu verstehen, vorherzusagen und zu verändern (Schlagenhauser u. Amberg, 2014, S. 8). Die Theorie des geplanten Verhaltens ist als eine Revision von Ajzens *Theorie des überlegten Handelns* zu sehen, mit dem Unterschied, dass Erstere als zusätzliche Determinante berücksichtigt, ob eine Person sich in der Lage fühlt, ihr Verhalten kontrollieren zu können (Ajzen, 1991, S. 181). Die Abbildung 17 zeigt die Theorie des geplanten Verhaltens und dessen Determinanten nach der Auffassung des Begründers Ajzen. Wie in der Abbildung zu erkennen ist, ist das Verhalten einer Person abhängig von deren Intention, welche die motivationalen Faktoren beinhaltet, die zu einem bestimmten Verhalten führen (Ajzen, 1991, S. 181).

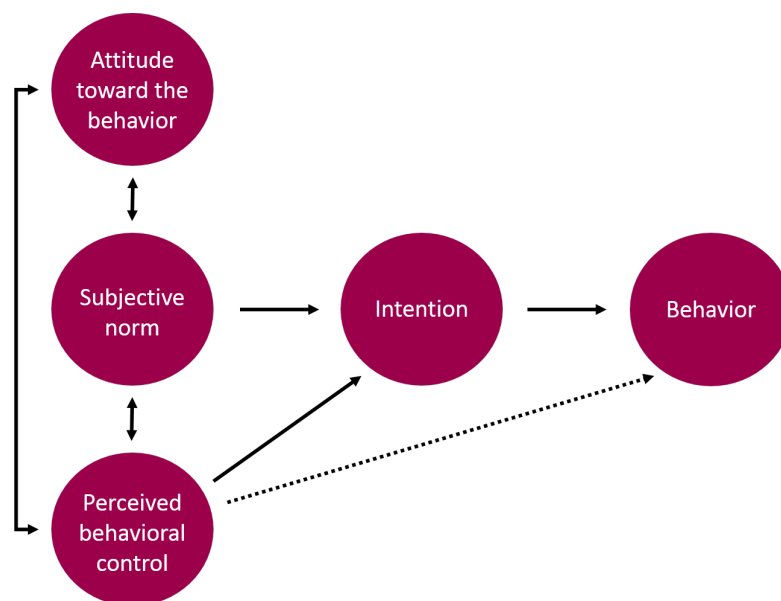


Abbildung 17: Theory of Planned Behavior (eigene Darstellung nach Ajzen (1991, S. 182))

Das wiederum ist das Ergebnis aus der subjektiven Einstellung gegenüber einem Verhalten (*attitude toward the behavior*), der subjektiven (sozialen) Normen (*subjective norm*) und der wahrgenommenen Verhaltenskontrolle (*perceived behavioral control*) (Ajzen, 1991, S. 181f.). Letztere entspricht der bereits vorgestellten Selbstwirksamkeitserwartung von Bandura (1977). Jedoch gibt Ajzen zu bedenken, dass eine starke Intention alleine nicht ausreichend ist, um ein Verhalten auszuführen (Ajzen, 1991, S. 181f.). Vielmehr muss die aus-



führende Person auch über ein ausreichendes Maß an *Volition* verfügen, also die Willenskraft besitzen, die eigenen Fähigkeiten so zu nutzen, dass die Intention tatsächlich zu einem Ergebnis im Sinne eines Verhaltens führt (Ajzen, 1991, S. 182).

Schlagenhauser u. Amberg sehen in der Theorie des geplanten Verhaltens eine wertvolle Möglichkeit für Gamification soziale Faktoren offenzulegen, welche Einstellungen von Individuen vorhersagen lassen und dabei Verhalten gezielt steuern können (Schlagenhauser u. Amberg, 2014, S. 8). Unabhängig aber dennoch fast zeitgleich zu der These von Schlagenhauser u. Amberg erkennen Conor et al. ebenso die Wichtigkeit, welche diese Theorie für den Gamification-Ansatz haben kann und veröffentlichen einen Vorschlag, wie eine Verhaltenssteuerung gezielt initiiert und durchgeführt werden kann. Sie postulieren hierzu ein Modell aus der angewandten Verhaltensanalyse (siehe Abbildung 18), welches ihrer Meinung nach die wichtigsten Prozesse bei Verhaltensinterventionen abbildet und sich damit – wenn auch nicht von den Autoren vorgesehen – an das Modell von Ajzen anlehnt.

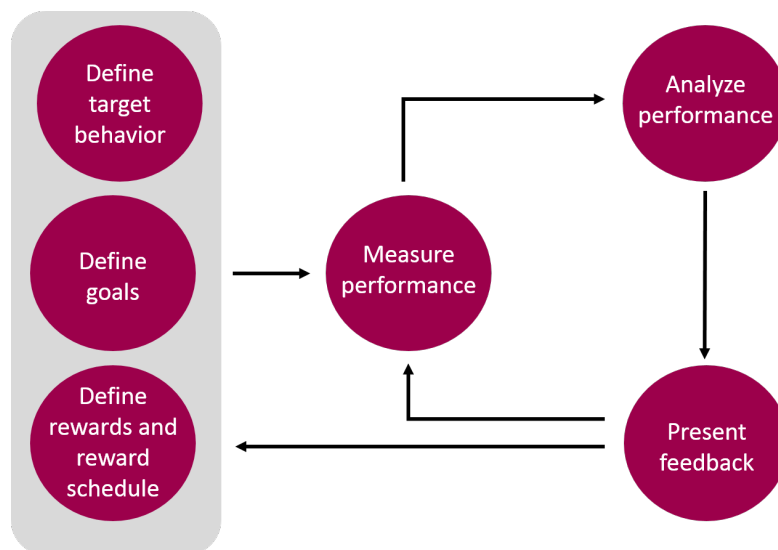


Abbildung 18: Modell der Schlüsselprozesse bei Verhaltensinterventionen (eigene Darstellung nach Conor et al. (2014, S. 97))

Ihr Modell sieht vor, dass zunächst eine Auswahl und möglichst genaue Definition des Zielverhaltens zu erfolgen hat. Dies sollte „[...] a clearly, objectively observable behavior [...]“ sein, welches messbar durch die Technologie ist, mit der ein System implementiert ist (Conor et al., 2014, S. 97). Die Beurteilung, ob eine teilnehmende Person das gewünschte Zielverhalten zeigt, soll dabei nicht durch Antworten auf Selbsteinschätzungsfragen erfolgen, sondern durch das System (Conor et al., 2014, S. 97). Hierzu muss ein Designer

eines solchen spielifizierten Systems darauf achten, dass kleine sowie beobachtbare und für Nutzer transparente, sowie erreichbare Ziele existieren (Conor et al., 2014, S. 98) – eine Anforderung, die sich häufig in einschlägiger Literatur zum Design von spielifizierten Anwendungen oder Systemen wiederfindet (siehe z.B. Bartel et al. (2016); de Byl u. Hooper (2013); Landers et al. (2017); Nicholson (2012); van Roy u. Zaman (2017)). Dieses Zielverhalten wird numerischen Werten zugeordnet, um das beobachtbare Verhalten zu beurteilen. Dabei soll nach Conor et al. darauf geachtet werden, dass nicht nur ein Verhalten als Ganzes gemessen wird, sondern einzelne Zeitabschnitte während der Verhaltensausführung, da diese mehr Aussagekraft in Bezug auf die Erreichung des gewünschten Zielverhaltens bieten (Conor et al., 2014, S. 98). Wird ein intendiertes Zielverhalten durch eine teilnehmende Person erreicht, muss im Vorfeld durch Designer bereits klar definiert werden, welches Feedback (konsistent, angemessen und bedarfsorientiert) bzw. welche Belohnungen (der Performanz angemessen, positive oder negative Belohnungen sowie zeitlich unmittelbar nach einer Aktion) wann zu erwarten sind (Conor et al., 2014, S. 98f.). Welche zeitlichen Messpunkte für eine Beurteilung sinnvoll sind und welche Daten bei einer Messung erhoben werden, bedarf ebenso einer sorgfältigen Abwägung, wie die Interpretation dieser Daten. Typischerweise, so Conor et al., wird jede einzelne Instanz, welche das Zielverhalten zeigt oder dazu beiträgt es zu erreichen aufgezeichnet und fließt in die anschließende Interpretation der Daten ein (Conor et al., 2014, S. 98). Entsprechend des Interpretationsergebnisses, wird der teilnehmenden Person das bereits definierte Feedback (bzw. Belohnung) präsentiert und dabei weitere Daten erhoben, die einen Abgleich zwischen gezeigtem Verhalten und Zielverhalten erlauben.

#### 3.2.4.5 *Uses and Gratification Theory nach Katz et al.*

Die letzte durch Schlagenhauser u. Amberg für Gamification als wichtig erachtete Theorie ist der *Nutzen- und Belohnungsansatz* (Uses and Gratification Theory) nach Katz et al. (1973). Es handelt sich dabei um ein aus der Meinungsforschung stammendes Modell, bei dem der Konsument von Medien als ein aktiver und bewusster Nutzer dieser verstanden wird (Katz et al., 1973, S. 510f.). Die Theorie findet ihren Ursprung in der empirischen Massenmedienforschung zu Beginn der 40er Jahre des letzten Jahrhunderts (Katz et al., 1973, S. 509). Die Theorie besagt, dass die Motive und Interessen einer Person bestimmen, in welchem Ausmaß und zu welchem Zweck Medien (z.B. Zeitung, Radio, Fernsehen, Internet) genutzt werden, um ihre Bedürfnisse zu erfüllen und bestimmte Ziele zu erreichen (Katz et al., 1973, S. 509ff.). Motive können dabei beispielsweise sein: Interaktionsersatz, Unterhaltung, Erholung,

Informations- und Wissenserwerb oder schlichtweg Zeitvertreib. Wichtig ist in diesem Zusammenhang zu erwähnen, dass Katz et al. mit dieser Theorie die bis dato vorherrschende Wahrnehmung des Konsumenten von Medien umdreht, hin zu einem aktiven Teilnehmer, der Medien als an ein Motiv geknüpft Mittel zum Zweck sieht, persönliche Bedürfnisse zu erfüllen (Katz et al., 1973, S. 513f.). Diese Bedürfnisse sind eng an die in dem Modell *Hierarchy of Needs*<sup>12</sup> von Maslow (2013) geknüpft. Neben den Motiven kommt es auf die selektiven Einflüsse an, die auf einen Menschen wirken bzw. von ihm ausgehen, um zu bestimmen, welches Medium für welche Zweckerfüllung dienlich ist (Katz et al., 1973, S. 512).

Aus Sicht von Gamification erscheint diese Theorie nach Auffassung von Kankanhalli et al. als sinnvoll, um die Motive zu erklären, warum Personen an Online Communities (z.B. Massive Open Online Courses (MOOC)) teilnehmen (Kankanhalli et al., 2012, S. 5). Dabei ist, wie oben bereits erwähnt, die *Nähe* eines Mediums zu persönlichen Interessen und die vorherrschenden jeweiligen Motive entscheidend bei der Auswahl eines Mediums.

Abschließend kann festgehalten werden, dass der Gamification-Ansatz aus motivationspsychologischer Sicht auf einer Vielzahl an verschiedenen Theorien aus der Verhaltensforschung, allgemeinen Motivationsforschung oder Sozialforschung aufbaut, die je nach Kontext und Ziel unterschiedlich<sup>13</sup> in die konkrete Ausprägung einer Spielifizierung einfließen können. Dabei sind die jeweiligen Theorien, wie bereits erwähnt, weder in jedem Fall trennscharf voneinander abzugrenzen, noch liegt sichtlich vergleichbaren Konzepten ein identisches Begriffsverständnis zu Grunde. Die unterschiedlichen motivationspsychologischen Konzepte sind dabei Nährboden für Forschungsarbeiten und gleichzeitig Strukturierungshilfe, um Kombinationen aus Game Design-Elementen und Motivationsstrategien zu designen, in einer spielifizierten Anwendung zu manifestieren und deren Wirken wissenschaftlich zu überprüfen (Kankanhalli et al., 2012, S. 7; Dichev et al., 2014, S. 95).

### 3.3 ZUR LERNMOTIVATION

Dass die Fähigkeit zu lernen für den Menschen eine unabdingbare Eigenschaft darstellt, steht wissenschaftlich gesehen außer Frage, denn der Mensch, so

<sup>12</sup> Für eine ausführliche Darstellung dieser Theorie unter Bezugnahme auf Gamification, sei an dieser Stelle auf Dichev et al. (2014, S. 82f.) verwiesen.

<sup>13</sup> Beispielsweise abhängig von den Adressaten einer spielifizierten Anwendung, welche anhand unterschiedlicher Spielertypen nach Bartle (1996) in *Achiever*, *Explorer*, *Socializer* oder *Killer* unterschieden werden können.

Schröder, „[...] ist aufgrund seiner mangelhaften Instinktausstattung besonders auf Lernen angewiesen“ (Schröder, 2002, S. 13). Der Begriff Lernen wird dabei von Schröder verstanden als das Bewirken einer „[...] relativ dauerhaften Verhaltensänderung aufgrund von Erfahrung“ (Schröder, 2002, S. 14), durch die „[...] Aufnahme, Verarbeitung und Speicherung von Informationen“ (Schröder, 2002, S. 15). Die Bereitschaft zu lernen, also das Vorhandensein von Lernmotivation, korreliert dabei stark mit dem Leistungsverhalten (Mietzel, 2007, S. 349). Interessant ist dabei die Tatsache, dass nicht nur die Quantität eines Leistungsverhaltens beeinflusst wird, also beispielsweise wie viele Differentialgleichungen ein Student löst, sondern auch dessen Qualität, also ob beim Lösen höhere kognitive Funktionen eingesetzt werden, welche dazu führen, dass Informationen besser verarbeitet und behalten werden (Mietzel, 2007, S. 349). Nun stellt sich die Frage, was Lernmotivation ausmacht und vor allem, wie diese gezielt gefördert werden kann. Diesen beiden Fragestellungen sollen im Folgenden beleuchtet werden.

### 3.3.1 *Lernmotivation und deren Bedeutung*

Folgt man Knörzer, so wird unter Lernmotivation die Bereitschaft eines Individuums verstanden, sensorische, kognitive und motorische Funktionen in einer vorstrukturierten Lernsituation darauf zu richten und derart zu koordinieren, dass ein klar definiertes Lernziel erreicht wird (Knörzer, 1976, S. 139). Krapp vertritt eine allgemeinere Definition und versteht unter dem Begriff der Lernmotivation „[...] jene Strukturen und Prozesse, die das Zustandekommen und die Effekte des Lernens bzw. einer Lernhandlung erklären“ (Krapp, 1993, S. 188). Rheinberg u. Fries stellen den Aspekt des bewussten und freiwilligen Handelns stärker heraus und definieren Lernmotivation als die „Bereitschaft einer Person zu all solchen Tätigkeiten, deren Ausführung einem Lernzuwachs förderlich sind, gleichgültig, ob die Person diesen Lernzuwachs beabsichtigt hat oder nicht“ (Rheinberg, 1986, S. 360). Bei dem Begriff handelt es sich also um einen Sammelbegriff der Motivationsarten, individuelle Interessen, Ziele- und Wertvorstellungen sowie das Leistungsmotiv beschreibt und sich damit – ähnlich wie der Motivationsbegriff – aufgrund seines Facettenreichtums nur indirekt fassen lässt (Spinath, 2011, S. 46). Erwähnenswert ist zudem, dass die Aussicht auf einen Kompetenzzuwachs oder einen Kompetenzgewinn, wie es Rheinberg u. Fries betiteln, Anreiz genug sein kann, Lernaktivitäten auszuführen (Rheinberg u. Fries, 1998, S. 169). Damit einhergehend sehen sie weitere, vornehmlich extrinsische Anreize, für die Steigerung von Lernmotivation, eine Folge des Kompetenzgewinns, wozu beispielsweise

„Anerkennung von anderen, symbolische oder materielle Gratifikation, Eröffnung weiterer Lernchancen [oder] einer Sache [bzw.] Idee dienen zu können“ (Rheinberg u. Fries, 1998, S. 169) gehören. In den genannten Anreizen manifestieren sich die Bestandteile der bereits erwähnten SDT, wie beispielsweise sich das Bedürfnis nach sozialer Zugehörigkeit durch Anerkennung durch andere erleben lässt.

### 3.3.2 Ein Rahmenmodell zur Lernmotivation nach Rheinberg und Fries

Aus Prozesssicht stellen Rheinberg u. Fries ein Rahmenmodell vor, welches die Bedingungen und Auswirkungen von Lernmotivation aufzeigt (siehe Abbildung 19).

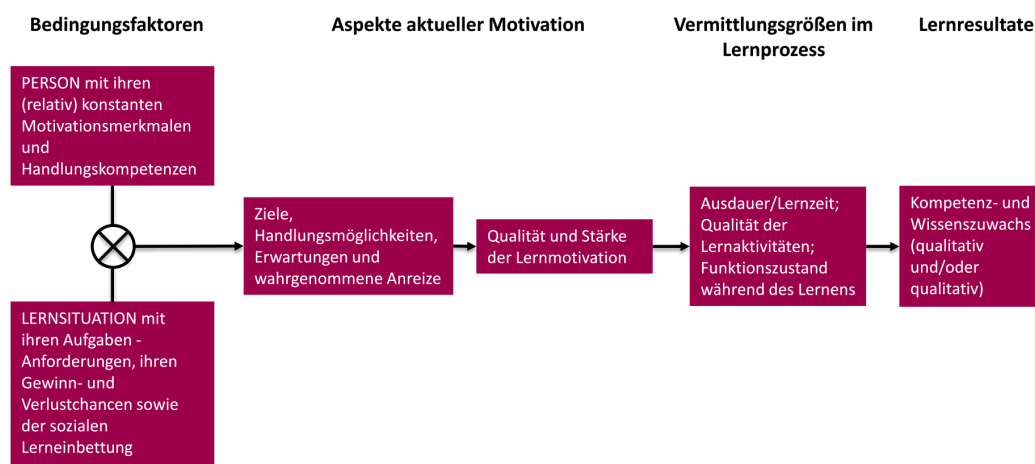


Abbildung 19: Rahmenmodell zur Lernmotivation und deren Auswirkungen nach (Rheinberg, 1986, S. 170)

Es unterscheidet zwischen Bedingungsfaktoren, welche die Person und ihre Motivationsmerkmale (kompetenz- und leistungsbezogenen Motive (Bandura, 1977; Heckhausen u. Gollwitzer, 1987) und überdauernde individuelle Interessen<sup>14</sup> (Krapp, 1993)) sowie Handlungskompetenzen betreffen und deren aktuelle Lernsituation, welche sowohl die Aufgabenanforderungen als auch die diesbezüglich empfundene Selbstwirksamkeit umfasst. Handlungskompetenzen lassen sich Rheinberg u. Fries zu Folge weiterhin unterteilen in volitionale (Heckhausen u. Gollwitzer, 1987) und kognitive (Weinstein u. Mayer, 1986) Handlungskompetenzen. Während erstere sich vor allem dann

<sup>14</sup> Rheinberg u. Fries geben an, dass Interessen sowohl einen Wertaspekt als auch einen emotionalen Aspekt besitzen. Wenn sich Interessen mit einem Lerngegenstand decken, weil zum Beispiel Lernenden Wahlfreiheit zugesprochen wurde, so kann sich dies positiv auf die personellen Motivationsmerkmale auswirken (Rheinberg u. Fries, 1998, S. 171). Infolgedessen ist eine positive Beeinflussung der Lernmotivation wahrscheinlich.

zeigt, wenn Lernende merken, dass ein Lernzuwachs wichtig wäre, sich aber weniger fordernde Handlungsalternativen anbieten, denen widerstanden werden will, ist zweitere bezogen auf „[...] lernspezifische Dinge wie allgemeine Arbeitstechniken [...], metakognitives Wissen [...], aufgabenspezifische Lernstrategien [und] aufgabenspezifische[s] Vorwissen und die Fertigkeit es einzusetzen (Rheinberg u. Fries, 1998, S. 172). Vor allem bei einer Einschätzung der Handlungskompetenzen spielt die Selbstwirksamkeitserwartung (Bandura, 1977) für Lernende in Bezug zu einer lernpädagogischen Erwartung eine maßgebliche Rolle. Denn wenn Lernende glauben Handlungskompetenzen erfolgreich einsetzen zu können, werden sie dies in aller Regel auch eher tun, so Zimmerman (2000).

Rheinberg u. Fries geben weiter zu bedenken, dass Lernmotivation nicht alleine durch Personenmerkmale bedingt wird. Vielmehr stehen diese in Wechselwirkung mit der jeweiligen Lernsituation. Darunter verstehen die Forscher die Anforderungen durch die Aufgaben, deren Gewinn- und Verlustchancen sowie die Art der sozialen Einbettung – Erkenntnisse, die sich aus Sicht von Studierenden auch mit aktueller Forschung<sup>15</sup> knapp 20 Jahre später decken (Figas u. Hagel, 2017). Deshalb ist es beispielsweise für das Zeigen von Interesse wichtig, ob „[...] die Lernsituation Aufgaben bietet, die aus einem interessenpassenden Gebiet stammen oder nicht“ (Rheinberg u. Fries, 1998, S. 172). Bezogen auf die Motive die Handlungskompetenz betreffend, ist es für die Lernmotivation förderlich, wenn die gestellte Aufgabe als Möglichkeit der Kompetenzsteigerung oder wenigstens der eigenen Kompetenzdiagnose angesehen wird (Rheinberg u. Fries, 1998). Auch wenn eine derartig charakterisierte Situation nicht besteht, so können dennoch vorwiegend extrinsische Anreize Lernprozesse initiieren, beispielsweise von Mitlernenden entsprechende Anerkennung erfahren zu können. Dies stellt damit Gewinnchancen für Lernende dar. Rheinberg u. Fries attestieren der Nützlichkeit bzw. Übertragbarkeit des Lernens ebenso eine große Wichtigkeit in Bezug auf die Lernmotivation. Diese Nennung deckt sich aus Lernendensicht erneut mit aktueller Forschung (Figas u. Hagel, 2017, S. 680).

Die aus Persönlichkeits- und Situationsmerkmalen entstehenden Motivations Tendenzen beeinflussen die Ziele einer lernenden Person, die Handlungsmöglichkeiten sowie die wahrgenommenen Anreize. Ziele können selbst- oder fremdbestimmt sein, wobei eine klare Zielformulierung einer Aufgabe einen positiven Einfluss auf die Lernmotivation haben kann (Figas u. Hagel, 2017).

<sup>15</sup> Figas u. Hagel (2017) fanden in einer Studie heraus, dass die praktische Relevanz, die Schwierigkeit sowie die Relevanz für eine Prüfung einer Aufgabe zu den wichtigsten motivationalen Faktoren gehören, welche aus Sicht von Studierenden des Software Engineerings maßgeblich für deren Lernmotivation ist (Figas u. Hagel, 2017, S. 680f.).

Erwähnenswert ist zudem, dass möglichst präzise formulierte Zielsetzungen, welche herausfordernd in deren Erreichung sind, die größten Anstrengungen und besten Leistungen bei Lernenden hervorbringen (Rheinberg u. Fries, 1998, S. 173). Das bewusste Wahrnehmen von Handlungsmöglichkeiten ist eine notwendige Voraussetzung für die Einschätzung realistischer Erfolgserwartungen. Jedoch ist nicht nur das Ergebnis<sup>16</sup> einer Handlungsmöglichkeit Anreiz für Lernende, vielmehr kann auch die dazu erforderliche Lerntätigkeit<sup>17</sup> selbst eine sein (Rheinberg u. Fries, 1998, S. 173). Beide Typen von Anreizen haben Einfluss auf die Lernmotivation und stehen beim institutionellen Lernen häufig im Gegensatz zueinander. So kann Lernenden durchaus bewusst sein, dass Lernaktivitäten einen hohen Grad an Folgeanreiz haben (z.B. eine gute Note oder schlicht einen Kompetenzzuwachs), dafür jedoch ein hohes Maß an Lerntätigkeit notwendig ist, weshalb der Tätigkeitsanreiz gering ist. In einem solchen Fall ist es möglich durch willensgesteuertes Lernen (Rheinberg u. Fries, 1998), also weniger effizientes und zugleich anstrengenderes Lernen dennoch Lerntätigkeiten auszuführen (Rheinberg u. Fries, 1998, S. 172f.). Erwartungen und Anreize werden abhängig von der persönlichen Einstellung, im Speziellen, „[...] der pro- bzw. retropektiven Ursachenklärung möglicher Leistungsresultate“ (Rheinberg u. Fries, 1998, S. 173) hin beurteilt. Rheinberg u. Fries geben hierbei die Metapher der individuell eingefärbten Brille an und spielen damit auf die Hervorhebung oder Ignoranz einzelner Anreize an. Dies erklärt unter anderem die Tatsache, warum sich manche Lernende durch bestimmte Anreize, wie beispielsweise die beste Zensur einer Gruppe zu erzielen, motivieren lassen und andere nicht.

Das Ergebnis dieser Determinanten ist die Qualität und Stärke der Lernmotivation von Lernenden. Die Stärke gibt dabei an, „[...] wie gut sich bei einer passenden Lerngelegenheit Lernaktivitäten gegenüber konkurrierenden Aktivitäten durchsetzen und wie sehr dann innere und äußere Prozesse in Richtung der Lernaufgabe und des Lernziels angeregt und fokussiert werden“ (Rheinberg u. Fries, 1998, S. 173). Die Qualität hingegen beschreibt die Güte der vorherrschenden Lernmotivation, welche für komplexe Aufgaben durch die Faktoren Erfolgszuversicht, Mißerfolgsbefürchtungen, Herausforderung und Interesse operationalisiert werden kann (Rheinberg u. Fries, 1998, S. 173).

Weiterhin sind für die Lernmotivation diverse Vermittlungsgrößen während des Lernprozesses von Bedeutung. Bisherige Studien konnten nur eine vergleichsweise geringe Anzahl an möglichen Einflüssen identifizieren, welche sich reproduzierbar auf die Lernleistung auswirken (siehe z.B. Schiefele

16 Auch bezeichnet als Zweckanreiz/Folgeanreiz (Rheinberg, 1989).

17 In diesem Fall spricht Rheinberg von sogenannten Tätigkeitsanreizen (Rheinberg, 1989).

(1996) als Studie zum Textlernen). Rheinberg et al. (2000) gehen jedoch davon aus, dass drei Wirkungsvariablen existieren, nämlich

- die Zeit<sup>18</sup>, die für die Lernaktivität in Anspruch genommen wird (auch die Häufigkeit der Zeiteinheiten),
- die Qualität der ausgeübten Lernaktivität (werden beispielsweise bestimmte kognitive Lernstrategien zum Lernen eingesetzt) sowie
- den motivational-funktionalen Zustand während der Lernaktivität (z.B. Aktivierung oder Konzentration) (Rheinberg u. Fries, 1998, S. 174).

Bezogen auf das Lernergebnis bzw. Lernresultat kommt es darauf an, wie die Lernleistung bewertet wird, denn davon abhängig treten Motivationseinflüsse zum Vorschein. Rheinberg u. Fries nehmen an, dass eine Wechselwirkung zwischen Quantität und Qualität eines Lernergebnisses besteht und folgen damit der Argumentation von Schneider u. Kreuz, welche sich für einen nicht unumstrittenen Mengen-Gute-Austausch aussprechen, den sie wie folgt verstehen (Schneider u. Kreuz, 1979; Rheinberg u. Fries, 1998): Eine Steigerung der Lernmotivation hat eine Steigerung der Bearbeitungsgeschwindigkeit zur Folge, was sich positiv in der Quantität der Leistung niederschlägt. Dies bedeutet, dass Lernende, die motivationsbedingt beispielsweise viele Aufgaben bearbeiten demnach einen höheren Output erzeugen, im Vergleich zu weniger motivierten Lernenden. Umgekehrt kann jedoch eine Quantitätssteigerung zu einem Abfall der Leistungsqualität führen (Schneider u. Kreuz, 1979). Ein Beispiel hierfür wäre eine Prüfungssituation, in der aufgrund der festgelegten und zu erbringenden Leistung Flüchtigkeitsfehler gemacht werden, die unter anderen Umständen vielleicht hätten vermieden werden können. Weiterhin ist für das Lernergebnis das Bezugssystem von essentieller Bedeutung, also der Kontext in dem ein Lernergebnis bewertet wird (Rheinberg, 1989, S. 174). Vor allem die Verstetigung einer Lernmotivationssteigerung hängt vom Bezugssystem ab, da dieses Lernzuwächse aufzeigt und für Lernende (vergleichbar) sichtbar macht. Das Bezugssystem kann dabei unterschiedlich ausgestaltet sein und beispielsweise einen sozialen Vergleich anstreben, also Lernenden miteinander direkt vergleichen, oder auf einzelne lernende Personen ausgerichtet sein und damit eine Eigenentwicklung pro Zeiteinheit aufzeigen.

Das von Rheinberg u. Fries (1998) entwickelte Rahmenmodell zeigt deutlich, dass Lernmotivation ein komplexes und vielfältiges Gefüge darstellt, welches teilweise nur indirekt von außen fassbar gemacht werden kann. Es bildet

<sup>18</sup> Siehe hierzu auch die Arbeit von Bloom (1984), welcher herausfand, dass die *Time on Task* stark mit dem gesamten Lernergebnis von Lernende korreliert.



jedoch ein maßgebliches Verständnis darüber, wie Motivationsprozesse beim Lernen funktionieren und vor allen Dingen, inwiefern sich eine Lernmotivationssteigerung bei Lernenden durch gezielte Interventionen realisieren lässt. Deshalb soll es im Folgenden um die Frage gehen, wie Lernmotivation gefördert werden kann, bzw. welche konkreten Maßnahmen dafür notwendig und sinnvoll erscheinen.

### 3.3.3 *Förderung von Lernmotivation*

Die Lernmotivation stellt einen zentralen, wenn nicht sogar den zentralen Antrieb für den Kompetenzerwerb und darüber hinaus für dessen Aufrechterhaltung dar (Smolka, 2004, S. IX). Bleibt dieser innere Antrieb aus, so können folglich keine Lernprozesse stattfinden und starke Diskrepanzen können zwischen Lernpotential und Lernerfolg eines Individuums auftreten (Dresel, 2010; Ulich, 1996). Dieser Zusammenhang erklärt unter anderem auch, warum es hochbegabte Menschen gibt, welche in pädagogischen Kontexten schwache Leistungen zeigen oder weniger Begabte zu den Besten einer institutionellen Lerngruppe zählen. Die Wichtigkeit der Lernmotivation manifestiert sich ebenso in der Formulierung von Dresel u. Lämmle, welche diese – neben der fachlich-methodischen Ausbildung – als „[...] wichtiges Bildungsziel jenseits ihrer Relevanz für schulische Leistungen“ (Dresel u. Lämmle, 2011, S. 84) sehen. Diese Aussage der Forscher lässt sich ohne Einschränkungen ebenso auf das hochschulische Umfeld übertragen. Die Förderung der Lernmotivation ist demnach in pädagogischen Kontexten ein wichtiges und zugleich notwendiges Ziel.

Für das Ausbleiben von Lernmotivation liefert die Motivationspsychologie zahlreiche Erklärungsmöglichkeiten. Erwähnenswert ist dabei, dass die Ursachen hierfür höchst individuell sind und meist nicht auf einzelne Faktoren reduziert werden können, wie beispielsweise der schlichte Mangel an fachlich-methodischem Wissen (nach Rheinberg u. Fries (1998) wären dies personenbezogene Handlungskompetenzen), um eine Aufgabe bewältigen zu können (Dresel, 2010). Vielmehr sind die Gründe auch weit darüber hinaus zu suchen, wie im familiären oder gesellschaftlichen Umfeld (nach Rheinberg u. Fries (1998) lernsituative Bedingungsfaktoren), in dem institutionelle Bildung keinen Einfluss mehr hat (Keller, 1995, S. 19). Demnach können andere Motivationen oder Kontextfaktoren existieren, welche die Lernmotivation überlagern können und dazu führen, dass diese geschwächt wird. Im Umkehrschluss bedeutet dies, dass Faktoren, die sich positiv auf die Lernmotivation auswir-

ken, sehr individuell sind und deren Ursache sowohl durch extrinsisches als auch intrinsisches motivationales Wirken bedingt sein kann. Das kann dazu führen, dass sich die Lernmotivation gegenüber anderen Motivationen durchsetzt (Spinath, 2011, S. 46). Entsprechend unterschieden werden können hierbei unterschiedliche Motivationsformen, wie Leistungsmotivation, Anschlußmotivation (Freude über eine soziale Beziehung) oder Wettbewerbsmotivation (Hoblitz, 2015a, S. 88; Mietzel, 2007, S. 363f.; Mazur, 2006, S. 427; Edelmann u. Wittmann, 2012, S. 231f.).

Neben der Lernendensicht ist ebenso die Lehrendensicht ausschlaggebend für die Lernmotivation von Lernenden. So können „[...] das Verhalten von Lehrenden, die Aufbereitung der Inhalte und die Gestaltung der Lernumgebung“ (Spinath, 2011, S. 45) eine starke Auswirkung auf die Lernmotivation haben. Dieser Umstand zeigt, dass ein singulärer Erkläransatz für ein bestimmtes Lernmotivationsniveau beim Lernen zu kurz greift (Figas et al., 2013). Es handelt sich dabei vielmehr um ein komplexes Gefüge, welches im Ergebnis auf vielfältige personale und situationale Ursachen zurückzuführen ist, die miteinander in Wechselwirkung stehen (Hoblitz, 2015a, S. 91).

Trotz dessen existieren zahlreiche, teils sehr allgemeine Empfehlungen dafür, wie Lernmotivation gefördert werden kann (siehe z.B. Dresel u. Lämmle (2011, S. 84f.); Dresel (2010, S. 131ff.); Mietzel (2007, S. 343ff., 648ff.); Edelmann u. Wittmann (2012, S. 77f.); Rheinberg u. Fries (1998, S. 176); Rheinberg u. Krug (2005); Brophy (2010)). Wenn man von Motivationsförderung spricht, intendiert der Begriff, dass dabei ein fremdgesteuerter automatischer Prozess bei einer Person in Gang gesetzt wird und diese zu einer Aktivität motiviert. Tatsächlich, und das zeigt unter anderem das Rahmenmodell von Rheinberg u. Fries (1998), ist das Wirkungsfeld von Personen, die andere zu einer Aktivität motivieren wollen, vergleichsweise gering. Dies scheint vor allem in lernpädagogischen Szenarien eine Herausforderung darzustellen, denn es ist weder möglich Lernende von außen zu motivieren, noch kann sich Lernmotivation angeeignet werden (Figas et al., 2013, S. 1301). Der Neurowissenschaftler und Psychologe Manfred Spitzer umschreibt diesen Umstand wie folgt:

„Geht man den Gründen für die Frage zur Motivationserzeugung nach, so stellt sich heraus, dass es letztlich um Probleme geht, die jemand damit hat, dass ein anderer nicht das tun will, was er selbst will, dass es der andere tut. [...] Jemand muss, so scheint es, einen anderen motivieren. Das ist etwa so, wie wenn man jemandem Hunger beibringen wollte“ (Spitzer, 2003, S. 192).

Folglich existieren vergleichsweise wenige Möglichkeiten, um direkt Einfluss auf die Lernmotivation von Lernenden zu nehmen (Dresel u. Lämmle, 2011). Eine Möglichkeit ist das direkte Ansprechen der motivationalen Persönlichkeitsmerkmale, die auch in der SDT zu finden sind, also die Bedürfnisbefriedigung von Autonomie, Kompetenzerfinden als auch von sozialer Eingebundenheit. Auf Basis dieser Erkenntnisse wurde in der Vergangenheit ein Modell entwickelt, welches sich für die Lehre des Software Engineerings an der Hochschule Kempten eignet und auf Basis dessen diverse Lehr-Lerneinheiten sowie Lernmaterialien konzipiert werden (Figas et al., 2013, S. 1301ff.). Es unterteilt in insgesamt vier Determinanten, die als lernmotivationsförderlich angesehen werden (siehe Abbildung 20).



Abbildung 20: Modell zur Förderung der Lernmotivation im Software Engineering nach (Figas et al., 2013, S. 1302, eigene Übersetzung)

#### 3.3.3.1 *Neugierde wecken*

Lernmaterialien in ein „Neugierdeproblem“ zu verwandeln, das ist laut Roth einer der vielversprechendsten Ansätze, um Anreize für Lernaktivitäten bei Lernenden zu schaffen (Roth, 1983, S. 42). Dabei kann es ausreichend sein, wie von Novak et al. vorgeschlagen zu Beginn einer Lerneinheit Stimuli bei Lernenden zu erzeugen, indem eine oder mehrere Warm-Up Aufgaben zur Bearbeitung bereitgestellt werden (Novak et al., 1999; Cashman u. Eschenbach, 2003). Berlyne spricht in diesem Zusammenhang von der Erzeugung sogenannter epistemischer Neugierde (Berlyne, 1960). Diese wird erregt, wenn „ein Mensch Informationen zur Kenntnis nimmt, die mit seinem Wissen, sei-

nen Überzeugungen oder Einstellungen nicht, oder nur teilweise zu vereinbaren sind“ (Mietzel, 2007, S. 358). Roth formuliert, dass

„[...] Antriebe wie Neugier zu Motivierung von Lernleistung völlig ausreichen, ja dass diese unter Umständen wirksamer sein können als die üblichen Laboratoriumsantriebe, das ist Hunger, Durst und Furcht“ (Roth, 1983, S. 229).

Diese Form der Neugierde<sup>19</sup> ist abzugrenzen von situativem Interesse (Hidi u. Anderson, 1992). Letzteres ist vielfach an einen bestimmten Inhalt geknüpft, welches ein längeres Interesse bei Lernenden erzeugt, als es ein kurzes Ereignis im Falle von Neugierde tun würde (Mietzel, 2007). Zur Erzeugung situativen Interesses kann es sich um oben genannte Aufgaben handeln, die in einen entsprechenden Erzählkontext (z.B. angereichert durch Praxiswissen) eingebettet<sup>20</sup> sein können. Auch kann der Lerninhalt per se für die Erzeugung situativen Interesses förderlich sein (Keller, 1995). Wenn die Verknüpfung von situativem Interesse (Hidi u. Anderson, 1992) und persönlichem Interesse (Rheinberg, 1989) gelingt, so ist nach dem Rahmenmodell von Rheinberg eine höhere Motivationstendenz zu erwarten.

### 3.3.3.2 *Unabhängigkeit fördern*

Dass die Autonomie zu den (universalen) Grundbedürfnissen eines Menschen zählt und damit motivationsförderlich wirken kann, zeigt die SDT (Deci u. Ryan, 1985). Autonomie kann dabei auf unterschiedlichen Ebenen in Lernaktivitäten eingebettet werden. So wurde basierend auf dem vorgestellten Modell nach Figas et al. Lernenden Wahlfreiheit in der Aufgabenbearbeitung, der dabei vorherrschenden Sozialform sowie der Projektthemas bei der Durchführung von Projektarbeiten gegeben (Figas et al., 2013). Obwohl hierzu keine wissenschaftliche Validierung erfolgte, war eine höhere Identifikation der Studierenden mit den einzelnen Lernaktivitäten zu beobachten, was zu einer wahrnehmbaren Lernmotivationssteigerung führte und nach wie vor führt (Figas et al., 2013, S. 1303). Diese gemachten Beobachtungen und die darunterliegenden kausalen Zusammenhänge zwischen Autonomiegefühl (bzw. Kontrollgefühl) und Lernmotivationssteigerung sind aber bereits an anderer Stelle nachgewiesen worden (siehe z. B. Deci u. Ryan (1985); Mietzel (2007); Rhein-

19 Edelmann u. Wittmann sprechen hierbei auch von Neugiermotivation und verstehen darunter ein Mensch und Tier angeborenes permanentes Explorationsbedürfnis (Edelmann u. Wittmann, 2012, S. 230).

20 Hier sei das Konzept des Szenario-basierten Lernens erwähnt und für eine detaillierte Beschreibung auf Bartel et al. (2014b) verwiesen.

berg u. Fries (1998)).

### 3.3.3.3 *Schwierigkeit individuell anpassen*

Bei der dritten Determinante steht die Anpassung der Schwierigkeit im Vordergrund. Diese erfolgt unter der Annahme, dass die Lernmotivation von Lernenden dann gefördert werden kann, wenn das Lernniveau bereits vorhandenes Wissen aufgreift und daran anknüpft, keine Über- oder Unterforderung für Lernende gegeben ist und gleichzeitig etwas Neues für Lernende darstellt (Figas et al., 2013, S. 1303). Diese Sichtweise ist eng mit der Selbstwirksamkeitstheorie (Bandura, 1977) verbunden und legt zu Grunde, dass „nur wenn Lernende sich selbst als effizient und erfolgreich empfinden, können sie Gefallen am Lernen finden und intrinsische Motivation entwickeln“ (Figas et al., 2013, S. 1303, eigene Übersetzung). Folglich ist eine Differenzierung vonnöten, die beispielsweise auf Aufgabenebene nach dem Konzept unterschiedlicher Lernpfade<sup>21</sup> einzelne Bearbeitungsstränge anbietet, die sich in ihrem Lernniveau unterscheiden und bzw. oder im Sinne des Scaffolded Learning (z.B. (Renken et al., 2016)) Lernende schrittweise an das Lösen von Aufgaben heranzuführt. Dies wird unter anderem der Heterogenität von Lernenden gerecht, die an deutschen Hochschulen vorzufinden ist (Abke et al., 2012b).

### 3.3.3.4 *Anreize schaffen*

Der vierte und letzte Bestandteil des Modells fokussiert die Schaffung von Anreizen für Lernende. Eine beliebte und zugleich vielfältig eingesetzte sowie erfolgreiche Möglichkeit hierfür ist der Einsatz von vorwiegend extrinsischen Mitteln der Spielifizierung (siehe z. B. Bartel et al. (2014a); Bartel u. Hagel (2014)). Basierend auf dem Modell konnte gezeigt werden, dass das Ansprechen von Emotionen zusammen mit einer positiven Verstärkung in Form von Belohnungen in einem spielerischen Kontext zu einer Lernmotivationssteigerung geführt hat (Bartel u. Hagel, 2016b). Den hierfür notwendigen Umsetzungsmöglichkeiten und theoretischen Grundlagen beim spielifizierten Lernen, mit dem Ziel einer Lernmotivationsförderung, soll sich im Folgenden gewidmet werden.

---

<sup>21</sup> Nach Roth (2015) existiert für den Begriff des Lernpfads keine einheitliche und anerkannte Definition. Vielmehr existieren eine Vielzahl an Eigenschaften, die einen Lernpfad ausmachen (Roth, 2015, S. 6).

### 3.3.4 Designprinzipien zur Förderung von Lernmotivation durch Gamification

Neben dem oben vorgestellten Modell zur Lernmotivationsförderung in nicht spielifizierten Lernkontexten existieren in der einschlägigen Literatur zahlreiche Designprinzipien, welche für *meaningful Gamification*, wie es Nicholson (2012) nennt, zu Rate gezogen werden können. Darunter versteht er die „[...] integration of user-centered game design elements into non-game contexts“ (Nicholson, 2012, S. 5), mit dem Grundgedanken „[...] that there are some activities people engage in because they have intrinsic or internalized motivations for doing so“ (Nicholson, 2015, S. 4). Im Folgenden sollen ausgewählte Prinzipien vorgestellt werden, die die höchste Übereinstimmung in einschlägiger Literatur vorweisen. Die Vorarbeit dazu beruht auf den Designempfehlungen zur Konzeption von spielifizierten Lernaufgaben in einem Blended-Learning Kontext von Bartel et al. (2016, S. 7), welche sich auf allgemeinere Designprinzipien abstrahieren lassen und damit das bereits vorgestellte Motivationsmodell ohne Spielifizierung um die folgenden Prinzipien ergänzen.

#### 3.3.4.1 Transparente Zielorientierung

Nicholson (2012) stellt mit seiner Definition von *meaningful Gamification*<sup>22</sup> den Benutzer in das Zentrum des Designs und fordert, dass bei jeder Designentscheidung die Frage nach dem Nutzen für den User im Vordergrund stehen soll und wie dieser Nutzen durch gezielte Information repräsentiert werden kann (Nicholson, 2012, S. 5). Der Nutzer soll in der Lage sein, die Verbindung von Game Design-Elementen zu nicht-spielerischen Aktivitäten, wie dem Lernen, zu erkennen und dabei seine eigenen Ziele und Wünsche wiederfinden (Nicholson, 2012, S. 5; Hamari u. Koivisto, 2013, S. 8). Nicholson unterstützt seine Argumentation mit dem Konzept des Universal Design for Learning (UDL) nach Rose u. Meyer (2002), wonach Lernende die Möglichkeit besitzen sollen, sich auf unterschiedliche Art und Weise Lerninhalte anzueignen und im Ergebnis ihre dabei erzielte *Mastery* ebenso heterogen zu demonstrieren (Rose u. Meyer, 2002). Dieses Zielverständnis wird auch von de Byl u. Hooper geteilt, welche *fokussierte Ziele*, also Ziele eines spielifizierten Systems, welche mit den persönlichen Zielen von Lernenden übereinstimmen, als ein wichtiges Kriterium für ein erfolgreiches Design ansehen (de Byl u. Hooper, 2013, S. 223). Ihre Argumentation deckt sich mit der von Landers et al. (2017), welche sich auf die Theorie von Doran (1981) stützen, wonach

<sup>22</sup> Der Begriff *meaningful* wird neben Nicholson auch von weiteren Autoren genutzt, beispielsweise Barata et al. (2013). Letztere legen ein vergleichbares Begriffsverständnis zu Grunde.

alle Ziele SMART<sup>23</sup> sein sollen, also specific (spezifisch), measurable (messbar), assignable (zuweisbar), realistic (realistisch) und time-related (an zeitliche Vorgaben gebunden) (Doran, 1981, S. 36). Zu der SMART-Aufzählung sollte nach Kapp et al. ein weiteres Attribut aufgenommen werden, nämlich Transparenz. Wenn keine ausreichende Transparenz auf die Ziele eines Systems oder einer Intervention bzw. eines Arrangements vorliegt, ist die Erwartungshaltung an Lernende nicht eindeutig kommuniziert und deren Lernqualität unter Umständen eingeschränkt (Ramirez u. Squire, 2014, S. 636): „Tell the learners what they will be learning and then ask them what they learned“ (Kapp et al., 2014, S. 18). Ebenso wäre ein persönlicher Kompetenzabgleich mit den gestellten Erwartungshaltungen durch fehlende Transparenz erschwert und würde tendenziell in einer Ablehnhaltung seitens der Lernenden resultieren (Bartel u. Hagel, 2016a). Demnach ergibt sich folgendes Mapping (siehe Tabelle 5) der SMART-Eigenschaften (Doran, 1981, S. 36) für den spielifizierten Lernkontext:

<b>Eigenschaft</b>	<b>Beschreibung</b>	<b>Mapping auf spielifiziertes Lernen</b>	<b>Beispiele</b>
Spezifisch (specific)	Adressiere einen spezifischen Bereich, der verbessert werden soll	Erzeuge explizite Lernziele	Konkrete (Template-gestützte) Lernzielbeschreibung
Messbar (measurable)	Quantifiziere oder nenne eine Fortschrittsmetrik	Quantitative oder qualitative Fortschrittsmetriken	Punkte, Badges, Levels etc.
Zuweisbar (assignable)	Lege fest, wer was tut.	Zielgruppe bzw. Rollen beim Lernprozess festlegen	Rollen, Avatare, Regeln, Zielgruppenanalyse
weiter auf nächster Seite			

<sup>23</sup> Dieses Akronym stammt ursprünglich aus dem klassischen Management, hat jedoch in der Vergangenheit auch in Lernkontexten Anwendung gefunden (siehe z. B. O'Neill et al. (2006)).

Eigenschaft	Beschreibung	Mapping auf spielifiziertes Lernen	Beispiele
Realistisch (realistic)	Annahmen treffen, welche Ziele realistisch mit gegebenen Ressourcen erreicht werden können.	Unter Berücksichtigung der Lernenden realistische und herausfordernde Erwartungen erzeugen und diese an vorhandenen Kompetenzen ausrichten/anknüpfen	Progression to Mastery (Zichermann u. Cunningham, 2011, S. 29ff.)
Zeitgebunden (time-related)	Lege fest, wann Ergebnisse erreicht werden können.	Optional Constraints an Ziel knüpfen	Countdown Timer, Deadlines etc.

Tabelle 5: SMART-Zieleigenschaften im Kontext spielifizierten Lernens

Nah et al. fügen außerdem hinzu, dass sie *layered Goals*, also Ziele, die sich auf unterschiedlichen Ebenen befinden, für motivationsförderlich halten (Nah et al., 2013, S. 100). Dabei sehen sie drei Arten, wie die Zielstrukturen zeitlich aufgebaut werden können: kurz-, mittel- und langfristige Ziele, die Lernende dazu ermutigen Mastery zu erlangen (Nah et al., 2013, S. 100). Auch Morford et al. (2014) befürworten die Existenz von *subgoals*, welche für Fortschritt bei der Zielerreichung eines übergeordneten Ziels sorgen (Morford et al., 2014, S. 28). Diese Form der Zielorientierung, welche auf Mastery<sup>24</sup> abzielt und gleichzeitig das Kompetenzerleben befriedigt, ist außerdem in zahlreichen weiteren Publikationen vergleichbar argumentiert und wird als ein essentieller Bestandteil erfolgreicher Motivationsförderung beim spielifizierten Lernen erachtet (z. B. in Werken von Deterding (2015), Ramirez u. Squire (2014), Landers et al. (2017), de Byl u. Hooper (2013), Kapp et al. (2014), Deterding (2013a), Bartel et al. (2016) und Zichermann u. Cunningham (2011)).

<sup>24</sup> Kapp (2012b) sieht neben Mastery Zielen noch eine weitere Zielausprägung für weniger komplexe und vornehmlich repetitive Lernaktivitäten, die Performance Ziele (Kapp, 2012b, S. 224f.). Über ein allgemein geeignetes Verhältnis beider Zielarten in einer spielifizierten Lernumgebung wird jedoch keine Angabe gemacht.



### 3.3.4.2 Individuelle Lernpfade

Bei diesem Punkt wird im Vergleich zum vorgestellten Motivationsmodell eine Substitution für die Determinante *Schwierigkeit individuell anpassen* vorgenommen. Die Schwierigkeit von Lerninhalten für Lernende individuell anpassen zu können, impliziert, dass Lernende die Möglichkeit haben, selbstgesteuert zu wählen, welche Schwierigkeit ihrem persönlichen Kompetenzerleben entspricht und bei welchem Schwierigkeitslevel sie ein Kompetenzerleben erwarten (Rheinberg u. Fries, 1998; Weinert, 1999; Figas et al., 2013; Lee u. Hammer, 2011; Deterding, 2015; Simões et al., 2013). Die Nutzung dieses individuellen Lernangebots durch Lernende resultiert in individuellen Lernpfaden welche beim spielifizierten Lernen idealerweise einer nicht-linearen Storyline folgen (Liu u. Peng, 2013). Damit wird die Entscheidungsfreiheit in die Hand der Lernenden gelegt und dem Autonomieempfinden, begründet durch die SDT, genüge getan (van Roy u. Zaman, 2017). Nach Kapp et al. können derartige Erzählungen Emotionen bei den Spielern erzeugen, welche in einem Lernkontext von nachhaltiger Bedeutung für die Einprägsamkeit und die Nachhaltigkeit des Erlernten sind (Kapp et al., 2014, S. 238)<sup>25</sup>. Die Anordnung einer Erzählung kann hierbei auf unterschiedliche Arten erfolgen<sup>26</sup>. In einem linearen und gerichteten Fall navigiert der Spieler entlang einer Erzählung (Schell, 2015, S. 369; Adams u. Rollings, 2007). Für den Spieler gibt es dabei nur einen definierten Anfang und ein definiertes Ende. Bei einer netzwerkartigen Anordnung werden einzelne Stationen in einer Erzählung definiert und über Pfade miteinander verbunden (Schell, 2015, S. 370; Ryan, 2001; Byrne, 2005; Adams u. Rollings, 2007). Somit können Stationen über mehr als einen Pfad erreicht werden. Sollen dennoch verpflichtende Lernaktivitäten integriert werden, so sollen diese „meaningful in regard to the main activity“ (Knaving et al., 2013, S. 133) sein, da diese sonst die Fokussierung von Lernenden hemmen und sich negativ auf deren Motivation auswirken können (Knaving et al., 2013, S. 133).

Eine entsprechend positive Auswirkung auf die Motivation konnte in Verbindung mit einem moderaten Autonomieempfinden bereits erfolgreich nachgewiesen werden (Rigby u. Ryan, 2011; Deterding, 2015). Zudem entspricht diese Wahlfreiheit dem Gedanken der UDL-Theorie, wie von Nicholson postuliert (Nicholson, 2015, S. 7) und unterstützt weiterhin einen konstruktivistischen Lerngedanken (Rose u. Meyer, 2002). Eine Herausforderung scheint zu

<sup>25</sup> Kapp spricht in diesem Zusammenhang auch von dem Erzeugen einer idealen Interessenskurve bei den Lernenden, basierend auf einer guten, individuell zu durchlaufenden Story. Dabei ist das Interesse zu Beginn und am Ende eines Lernprozesses am größten.

<sup>26</sup> Weitere Arten der Anordnung (z.B. offene Welten), außer den hier genannten, sind in Schell (2015) und (Dormans, 2012, S. 110 ff.) beschrieben.

sein, die einzelnen Lernaktivitäten so zu gestalten und zu strukturieren, dass sie „[...] zum einen maximale Freiheitsgrade für die Studierenden erlauben und zum anderen auch ein gemeinsames Arbeiten mit einem vergleichbaren Wissensstand gewährleisten“ (Bartel et al., 2016, S. 7). Es soll gelten, dass die Flexibilität des Lernweges nur „[...] durch einen produktiven Beitrag zur Lernzielerfüllung begrenzt [wird]“ (Bartel et al., 2017d, S. 2). Eine Überforderung durch die Auswahl an Möglichkeiten kann ebenso eintreten (van Roy u. Zaman, 2017), wobei jedoch ein vorgeschlagener Lernpfad, welcher eine Storyline in für den Nutzer erkennbare Quests oder Tasks unterteilt, für effektives Lernen und leichte Orientierung sorgen kann (Ramirez u. Squire, 2014, S. 645) (Niman, 2014, S. 91). Die Rolle von Lehrenden wandelt sich dabei von Bereitstellern von Lerninhalten hin zu beratenden Mentoren (Deterding, 2013a, S. 61).

#### 3.3.4.3 *Kontextbezogenes Feedback*

In diesem Punkt herrscht große Einigkeit, dass unmittelbares, individuelles und informationsreiches Feedback maßgeblich dazu beiträgt (Bartel et al., 2016, S. 7), Lernende innerhalb von spielifizierten Lernkontexten in sogenannten *Feedback-Loops* (Zichermann u. Cunningham, 2011) zu integrieren und zu halten (Nicholson, 2012). Unter Feedback-Loops verstehen Zichermann u. Cunningham essentielle Bestandteile von allen Spielen, welche häufig im Zusammenspiel zwischen Punkten und Leveln auftreten (Zichermann u. Cunningham, 2011, S. 77). Da sich Punkte über den zeitlichen Verlauf verändern, sorgt diese Veränderung für unmissverständliches, individuelles und klares Feedback, dass ein Spieler auf dem richtigen Weg sei (Zichermann u. Cunningham, 2011, S. 77) und hat damit einen direkten Einfluss auf den Zustand des Systems (Morford et al., 2014, S. 28). Die Forscher de Byl u. Hooper sehen im Feedback eine Art der zielgerichteten Kommunikation mit dem Nutzer eines spielifizierten Systems, in dem Status und Verhalten von Personen, welche mit dem System interagieren, durch das System gespiegelt werden (de Byl u. Hooper, 2013, S. 223). Der Informationsaspekt bei einem Feedback solle nicht nur dafür genutzt werden, einem Spieler mitzuteilen, dass es sich bei einer Aktion um eine positive Aktion handelt, indem ein Spieler beispielsweise eine Belohnung erhält. Vielmehr soll Feedback dafür genutzt werden, um zu verdeutlichen, warum es sich bei einer Aktivität um eine belohnungswürdige Aktivität handele (Nicholson, 2015, S. 8). Dies kann beispielsweise in Form einer Nachricht an den Nutzer umgesetzt werden, welche mit der Vergabe eines Badges einhergeht. Nah et al. ergänzen, dass sie sowohl positives Feedback durch Components, wie Punkte oder Badges als motivationsförder-

lich ansehen, aber auch negatives Feedback, welches Lernende beispielsweise korrigiert, für wichtig halten (Nah et al., 2013). Negatives Feedback, so Nah et al., trage dazu bei, Lernenden dabei zu helfen, Ihre Ziele schneller zu erreichen (Nah et al., 2013, S. 101). Andere Forscher, wie van Roy u. Zaman (2017), Simões et al. (2013), Herrlich et al. (2014) oder Niman (2014) lehnen negatives Feedback ab, da sie eine negative Beeinflussung der autonomen Motivation auf Basis der Erkenntnisse von Deci u. Ryan (2002) vermuten. Wie sich negatives bzw. positives Feedback nach Ansicht der Forschenden konkret ausgestaltet, darüber existieren keine Angaben. Es bleibt also die Frage im Raum, ob beispielsweise ein negatives Feedback mit einer Verringerung einer Punktzahl als eine Form der Sanktionierung einhergeht, oder ob schlicht ein textueller Hinweis auf ein verbesserungswürdiges Verhalten eine Form des negativen Feedbacks darstellt.

#### 3.3.4.4 *Anreizgestützte Fortschrittsüberwachung*

Während Feedback auf einem actio-reactio-Prinzip beruht, sorgt eine durch Anreize gestützte Überwachung des Lernfortschritts für eine Form des *Engagements*, welche nicht nur eine einzelne Feedback-Loop<sup>27</sup> betrachtet, sondern eine Aneinanderreihung von Feedback-Loops in einer Kette vorsieht:

„Feedback is about more than creating a single loop. Rather it is about constructing a series of loops that form a chain capable of providing support needed to continue making progress toward the completion of the game“ (Niman, 2014, S. 102).

Diese bereitgestellte Unterstützung, von der Niman spricht, sorgt in erster Linie dafür, dass positive Folgeanreize<sup>28</sup> geschaffen werden, damit sich Lernende in ihrem Lernprozess zielgerichtet weiterbewegen können (Hoblitz, 2015b). Diese Unterstützung durch Anreize wird von Dichev et al. als einer der einflussreichsten Faktoren zur Motivationsförderung gesehen (Dichev et al., 2014), sofern die eingesetzten Game Design-Elemente, wie beispielsweise Punkte oder Levels, an für Lernende erreichbare und begehrte Ziele geknüpft sind (Dichev et al., 2014, S. 90). Nach Stott u. Neustaedter (2013) setzt die Lehrform Just-in-Time Teaching (JiT) diese Förderung von Kontinuität besonders gut um, sofern zur Bewertung der einzelnen Leistungen von Lernenden Punkte eingesetzt werden, da diese die höchste Relevanz für Lernende besitzen würden (Stott u. Neustaedter, 2013, S. 2). Jedoch sollen nicht die Belohnungen an

27 Gelegentlich wird auch der Begriff Feedback-Cycle in der einschlägigen Literatur als synonym verwendet, wie z. B. zu finden in Simões et al. (2013).

28 Siehe hierzu auch das erweiterte kognitive Motivationsmodell von Rheinberg (2008, S. 131f.).

sich das Lernen motivieren, sondern vielmehr das gewünschte Lernziel das Lernen motivieren und derartige Game Design-Elemente nur unterstützend wirken, wie auch Kapp et al. postuliert:

„Those [game design] elements should be integrated into the solution, but should not drive the development. The desired learning outcome must drive the development“ (Kapp et al., 2014, S. 16).

Es zeichnet sich also eine Funktionsdualität des spielifizierten Systems ab, welche „[...] both directly support end user activity (by ease of use) and facilitate it through enjoyment and motivation“ (Deterding, 2015, S. 304) erfüllt. Auch Hoblitz erkennt diese Dualität und attestiert der Spielifizierung in formalen Lernkontexten eine Parallelität zweier Handlungsstrukturen: der Lern- und Spielhandlung (Hoblitz, 2015b, S. 90). Belohnungen dienen neben dem Anreizcharakter auch der relativen Positionsbestimmung im Vergleich mit Mitlernenden (Niman, 2014, S. 140; Dichev et al., 2014, S. 91). Ramirez u. Squire halten es ebenfalls für essentiell, dass die Annahmen seitens der Lehrenden, welche für den Einsatz von Fortschrittsmetriken getroffen wurden, den Lernenden transparent gemacht werden und idealerweise dafür sorgen, dass Lernende über deren Erreichung in Dialog miteinander treten (Ramirez u. Squire, 2014, S. 636f.). Eine zu umfangreiche Transparenz birgt jedoch die Gefahr, dass Lernende nur noch die Aktionen ausführen, von denen sie wissen, dass diese an Belohnungen geknüpft sind. In diesem Fall wäre die Motivation fast ausschließlich extrinsisch und deshalb nur von vergleichsweise kurzer Dauer. Iosup u. Epema (2014) sehen genau darin eine Ausnahme, indem beispielsweise ein Freischalten von zusätzlichen Lerninhalten als Belohnungen für Lernende verfügbar ist.

#### 3.3.4.5 *Freiheit, Fehler zu machen*

Die Freiheit für Lernende, während des Lernprozesses Fehler ohne Konsequenzen zu machen, ist Teil des konstruktivistischen Lerngedankens und wird als motivierender Faktor und Stilmittel der Spielifizierung gesehen: „The risk of failure without punishment is engaging“ (Kapp, 2012a, S. 66). Deterding bezeichnet das Fehler machen sogar als „[...] the central route to learning“ (Deterding, 2013a, S. 63). Spiele im Allgemeinen ermutigen Spieler dazu, Fehler zu machen, während sie die Spielwelt in einem geschützten Kontext explorieren und Grenzen der Spielwelt austesten (de Byl u. Hooper, 2013, S. 224; Ramirez u. Squire, 2014, S. 637f.; Simões et al., 2013, S. 348; Knaving et al., 2013, S. 133). Ähnlich argumentieren Stott u. Neustaedter (2013) und sehen dieses Prinzip vor allem in kombinierter Nutzung mit formative Assessment als eine effektive Art und Weise, um die Motivation von Lernenden zu erhöhen, da bei dieser

Lernform nicht das Endergebnis im Vordergrund steht, sondern vielmehr der Prozess währenddessen (Stott u. Neustaedter, 2013, S. 1). Dies bedeutet jedoch nicht, dass im Kontext eines formative Assessments Lernende zur korrekten Beantwortung jeder Aufgabe möglichst viele Chancen haben sollen, sondern vielmehr, dass seitens der Lernenden Risiken eingegangen werden (Ramirez u. Squire, 2014, S. 637; Kapp, 2012a, S. 66), welche durch Feedback-Loops unterstützt dazu anregen, Lerninhalte zu explorieren und eine positive Einstellung zum Begehen von Fehlern zu erzeugen (Lee u. Hammer, 2011, S. 3). Nach Kapp (2012a) impliziert dies ebenso, dass falsche Antworten bewusst aufgegriffen werden und dazugehörige Konsequenzen unter Annahme der falschen Antwort in das Feedback einfließen (Lernen aus Fehlern) (Kapp, 2012a, S. 67).

#### 3.3.4.6 Soziale Eingebundenheit und Vergleich

Ein weiteres vielversprechendes Prinzip für das Design von spielifizierten Lehr-Lernaktivitäten ist die Integration von Lernenden in soziale Netzwerke. Diese können nicht nur dazu dienen, sich mit Mitlernenden zu vergleichen und damit entsprechende Status von Lernenden gegenüber anderen zu repräsentieren (siehe z. B. Alves et al. (2014)), vielmehr sorgt eine soziale Eingebundenheit für die Befriedigung eines menschlichen Grundbedürfnisses, begründet durch die SDT. Unter Einbezug der Arbeit von Nah et al. (2013) geben Bartel u. Hagel an, dass Lernaktivitäten, welche so konstruiert sind, dass

„[...] sie entweder mit Hilfe einer sozialen Vereinigung [...] bearbeitet werden [können], oder diese Vereinigung direkt [betreffen], beispielsweise in dem das Ergebnis [einer Lernaktivität] einer anderen Vereinigung bekannt gemacht wird [...], kann dies die intrinsische sowie extrinsische Motivation von involvierten Lernenden steigern“ (Bartel u. Hagel, 2016b, S. 7).

Weiterhin wird dem Erfahrungsaustausch innerhalb einer Gruppe nachgesagt, dass dadurch das positive Gefühl sozialer Einbindung verstärkt werden kann (Rigby u. Ryan, 2011). van Roy u. Zaman formulieren adäquat hierzu:

„In an educational context, learners who work together, sharing experiences and a common goal, have stronger bonds, resulting in relatedness need satisfaction and autonomous motivation“ (van Roy u. Zaman, 2017, S. 499).

Dieser auf der SDT beruhende Effekt konnte bereits erfolgreich nachgewiesen werden. Die Forscher de Marcos et al. (2014) zeigten beispielsweise, dass die Integration von *social Gamification*, also die Einbettung von kommunikativen Elementen, zu höherer Lernperformanz führte. In einer späteren Studie verfeinerten sie Ihre Erkenntnisse und führten den Einsatz von *social Gamifi-*

cation darauf zurück, dass die Potentiale zur Wissensaneignung und Kompetenzentwicklung besser genutzt werden konnten (de Marcos et al., 2016). Hamari u. Koivisto konnten diese Ergebnisse bestätigen. Auch sie fanden in einer experimentellen Studie heraus, dass „[...] enabling users to get exposed to attitudes of others and also to receive feedback directly from other users can positively influence the attitude towards using a gamification service“ (Hamari u. Koivisto, 2013, S. 8). Sie halten soziale Elemente in spielifizierten Kontexten für essentiell, auch wenn diese zunächst soziale Normen in Bezug auf Herausforderungen verändern können, diese Dynamik jedoch auch gleichzeitig eine Chance dafür sein kann, die Stärke und Akzeptanz einer gemeinsamen Haltung gegenüber Herausforderungen zu erhöhen (Hamari u. Koivisto, 2013, S. 8). Von Bedeutung kann auch die soziale Anerkennung sein, die nicht nur innerhalb einer Gruppe von Individuen erfahren werden kann, sondern auch von Lehrenden als motivationsförderlich angesehen wird (Simões et al., 2013, S. 348).

All diesen vorgestellten Prinzipien ist gemein, dass sie ein hohes Maß an Kontextsensitivität besitzen, welche abhängig von der jeweiligen Intervention, unterschiedlich gut funktionieren, wie auch Stott u. Neustaedter feststellen:

„[...] [T]heoretical 'best practices' in the gamification of education are extremely context sensitive. There is no one-size-fits all model for the successful gamification of a classroom“ (Stott u. Neustaedter, 2013, S. 7).

Die Frage danach, was demnach erfolgreiche Spielifizierung ausmacht, beantworten Dicheva et al. eindeutig: „Successful gamification is not the one that utilizes most game elements but the one that utilizes game elements effectively“ (Dicheva et al., 2014, S. 2). Das zentrale Ziel der Lernmotivationsförderung bei Lernenden ist demnach mit einem äußerst heterogenen Einsatz von spielifizierten Elementen verknüpft. Diese Verbindung kann sehr unterschiedlich ausgeprägt sein, abhängig von der jeweiligen Mikrokultur, also der Lernkultur innerhalb einer Gruppe von Lernenden (Ramirez u. Squire, 2014, S. 645).

Die vorgestellten Prinzipien zum Design derartig spielifizierter Lehr-Lernarrangements ergänzen den bereits vorgestellten GCDP um Anregungen, in Form von Best Practices, wie Dynamics, durch darunterliegende Game Design-Elemente, operationalisiert werden können. Die Prinzipien schränken demnach auch die große Auswahl an Game Design-Elementen zur Operationalisierung der Dynamics entsprechend ein, auch wenn sie keine explizite Festle-

gung auf einzelne Elemente fordern, da es diese Verknüpfung noch zu validieren gilt (Dichev u. Dicheva, 2017; Deterding et al., 2013; Nacke u. Deterding, 2017):

„Additionally, success in one non-game context does not guarantee that the same mechanism will be successful in another non game context. Research is needed to describe the essential game mechanics in different contexts such as in the enterprise or in educational and learning environments“ (Richter et al., 2015, S. 37).

Die hier vorgestellten Best-Practices mündeten, unter Berücksichtigung der einschlägigen Literatur sowie der gesammelten Erfahrungen durch zahlreiche spielifizierte Lehr-Lernexperimente, in ein *Manifest für spielifizierte Hochschullehre*<sup>29</sup> (Bartel et al., 2017d).

### 3.4 ZWISCHENFAZIT UND FORSCHUNGSDESIDERATE

Trotz zahlreicher Motivationstheorien und erfolgreicher Adaptionen dieser Theorien in unterschiedlichste spielifizierte Lernkontexte bleibt die Frage offen, wie und ob verallgemeinerte und kontextunabhängige Regeln erarbeitet werden können, welche eine Abbildung zwischen „[...] psychological mediators and behavioural outcomes“ (Nacke u. Deterding, 2017, S. 452) zulassen. Damit wären Designaussagen möglich, welche „[...] long-term and in the wild, not just short-term and in the lab“ (Nacke u. Deterding, 2017, S. 452) Gültigkeit besäßen. Auch Dichev u. Dicheva (2017) bzw. Dichev et al. (2014) resümieren ähnlich zu dem aktuellen Stand der Forschung beim spielifizierten Lernen und kritisieren, dass die „[...] educational benefits of gamification in terms of increasing student motivation or linking this motivation to learning outcomes are still not well understood“ (Dichev u. Dicheva, 2017, S. 26). Sie sehen zudem die Kontextsensitivität eines jeden Einsatzes von Spielifizierung in Lehr-Lernkontexten als ein herausforderndes und bisher noch ungelöstes Problem (Dichev u. Dicheva, 2017, S. 25). Da die Forschenden unter spielifiziertem Lernen einen psychologisch getriebenen Ansatz sehen, der auf Motivation abzielt und aus technischer Perspektive ein motivationales Designproblem darstellt (Dichev u. Dicheva, 2017, S. 26), unterstellen sie, dass es unabdingbar ist, diese beiden Seiten weiter zu erforschen. Sie fordern zudem, dass die bisher unsystematische Art der Forschung beendet werden muss, bei der lediglich Game Design-Elemente bestehenden Lerninhalten hinzugefügt werden, um

<sup>29</sup> Der Beitrag von Bartel et al. (2017c) zeigt dabei nur die Andeutung des Entwicklungsvorhabens eines derartigen Manifests auf. Das vollständige Manifest ist zu finden unter: <http://gamefulmanifesto.org> und ist mit der Veröffentlichung des Beitrags entstanden.

vornehmlich extrinsische Anreize zu schaffen und Ziele von Lernenden zu ignorieren (Dichev et al., 2014, S. 93f.). Laut Nacke u. Deterding (2017) kann eine Harmonisierung und Standardisierung von spielifizierten Interventionen und Messwerkzeugen dazu beitragen, neue Erkenntnisse für die Forschung zu gewinnen (Nacke u. Deterding, 2017, S. 452). Dies würde die Vergleichbarkeit von Spielifizierungen steigern, Metaanalysen bzgl. deren behaviouristischen Effekten ermöglichen und damit dazu führen, dass systematisch neue Theorien entwickelt werden können (Nacke u. Deterding, 2017, S. 452). Dafür wird geeignete technologische Unterstützung benötigt, welche es erlaubt, die Spielifizierung von verschiedensten Kontexten effektiv zu unterstützen (Dicheva et al., 2015, S. 10).

Aus den durch die Forschenden geäußerten Aussagen ergeben sich Forschungsdesiderate, welche im Folgenden tiefergehend diskutiert werden sollen. Konkret soll die Idee der gesteigerten Standardisierung, welche bereits beim GCDP und dem damit verbundenen GCDR postuliert und angewandt wurde, weitere Forschungstätigkeiten motivieren. So soll im Folgenden der Arbeit eine mögliche Lösung basierend auf dem Ansatz des DSM vorgestellt werden, wie zunächst ein einheitliches und generalisiertes Grundverständnis von Game Design-Elementen und deren Einsatz in spielifizierten Lehr-Lernumgebungen geschaffen werden kann. Zunächst soll jedoch die Frage aufgegriffen werden, welche Game Design-Elemente sich typischerweise in Lehr-Lernkontexten wiederfinden und welches Verständnis sich über deren Struktur und Anwendung ausmachen lässt. Hierzu stellt das nächste Kapitel die sich daraus ergebenden Forschungsfragen, die Vorüberlegungen, die Durchführung sowie die Ergebnisse einer strukturierten Dokumentenanalyse und Lernplattformanalyse vor.



---

## EMPIRISCHE STUDIEN ZU GAMIFICATION IN DER LEHRE

---

*„Es ist nicht genug zu wissen; man muß es auch anwenden.  
Es ist nicht genug zu wollen; man muß es auch tun.“*

*— Johann Wolfgang von Goethe*

### 4.1 EINLEITUNG

Um das geforderte Maß an Standardisierung zu erhöhen, welches bei dem Design von spielifizierten Lehr-Lernarrangements von Nacke u. Deterding (2017) sowie von weiteren Forschenden gefordert wird, wird in vorliegender Arbeit zunächst ein Grundverständnis darüber gelegt, was die Spielifizierung strukturell auszeichnet und wie diese Strukturen in einem Lehr-Lernkontext voneinander abhängen, bzw. zusammen agieren. Hierfür ist es neben einem grundlegenden Verständnis der Domänenkonzepte der Spielifizierung in Lehr-Lernkontexten ebenso vonnöten, sich mit der Vielfalt dieser Konzepte auseinander zu setzen. Diese gilt es insoweit zu abstrahieren, als dass einzelne Konzepte im Kern immer gleich beschrieben und diese um (kontextspezifische) Variationen erweitert werden können. Die Erweiterung von derartigen Konzepten stellt eine technologische Herausforderung dar, für die der DSM-Ansatz jedoch Lösungsmöglichkeiten bietet. Das Exzerpieren von Konzepten und deren Kernbeschreibungen fordert hingegen eine analytische Tätigkeit, welche idealerweise im Sinne des DSR sowohl eine theoretische als auch praktische Sichtweise abdeckt. Aus diesem Grund wird im Folgenden die Durchführung einer Domänenanalyse beschrieben, welche sich aus vier aufeinander aufbauenden Analyseschritten zusammensetzt (siehe Abbildung 21).

In einem ersten Schritt wird ein Literature Review in Form einer Dokumentenanalyse durchgeführt, um aus wissenschaftlicher Literatur einen Theoriebasierten Exzerpt von Domänenkonzepten zu erhalten, die in spielifizierten Lehr-Lernarrangements zu identifizieren sind.

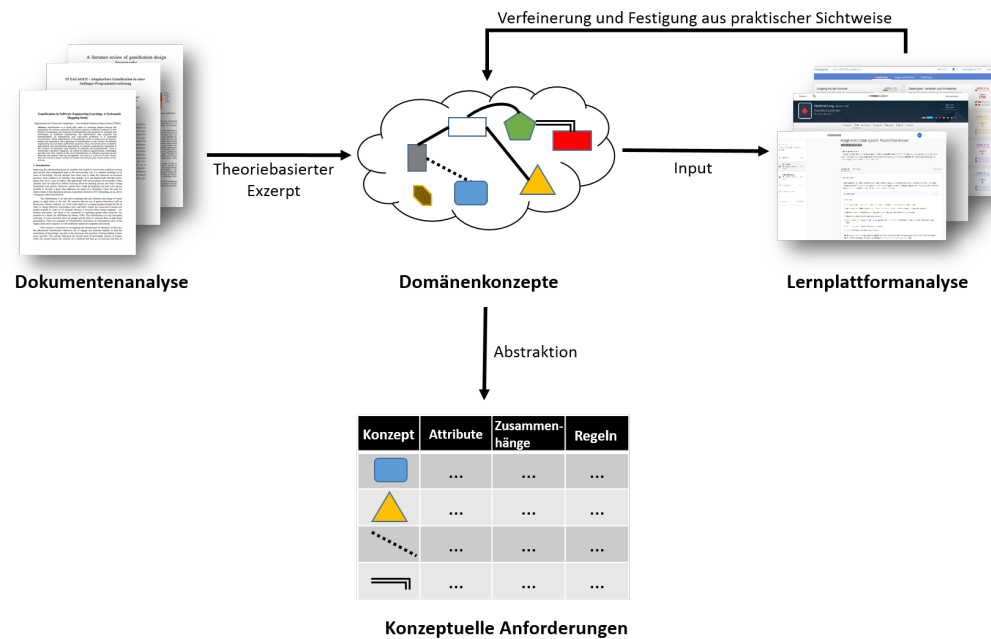


Abbildung 21: Schematischer Ablauf der Domänenanalyse

Die identifizierten Domänenkonzepte bilden den Input für eine praxisorientierte zweite Analyse, welche Lernplattformen auf Basis der Konzepte untersucht. Dies hat zum Ziel, dass die in der Dokumentenanalyse identifizierten Domänenkonzepte durch eine praktische Sichtweise gefestigt und in ihrer Detaillierung verfeinert werden können, was maßgeblich für den nächsten Schritt ist. Dieser besteht darin, dass die verfeinerten und gefestigten Domänenkonzepte anhand ihrer Attribute, Zusammenhänge untereinander und Regeln abstrahiert werden, um zu konzeptuellen Anforderungen zu gelangen, die für die anschließende Entwicklung einer domänenspezifischen Modellierungssprache verwendet werden.

Im Folgenden werden neben den theoretischen Grundlagen für die genannten Analyseschritte ebenso deren Durchführung und Ergebnisse geschildert, beginnend bei dem ersten Schritt, dem Literature Review in Form einer Dokumentenanalyse.

#### 4.2 DOKUMENTENANALYSE ZU GAMIFICATION DESIGN-ELEMENTEN IN DER LEHRE

Um den ersten Analyseschritt zu vollziehen, soll ein Literature Review durchgeführt werden. Für diesen Begriff existieren diskursive Definitionen, welche die Literaturlandschaft zu diesem Thema prägen. Beispielsweise definiert Fink ein Literature Review in der Forschung als

„[...] a systematic, explicit, and reproducible method for identifying, evaluating, and synthesizing the existing body of completed and recorded work produced by researchers, scholars and practioners“ (Fink, 2014, S. 3).

Obwohl ihr Buch die Domäne der Gesundheitswissenschaften fokussiert, gibt es durchaus Übereinstimmungen mit Definitionen aus anderen Disziplinen, wie den Informationswissenschaften. So verstehen Rousseau et al. unter einem (systematischen) Literature Review eine

„[...] comprehensive accumulation, transparent analysis, and reflective interpretation of all empirical studies pertinent to a specific question“ (Rousseau et al., 2008, S. 9).

Wie auch Fink sehen Rousseau et al. die Aufgabe eines Literature Reviews darin die Forschungsergebnisse in einem Forschungsfeld zu hinterfragen, Lücken zu erkennen sowie Grenzen und nicht validierte Annahmen aufzuzeigen (Rousseau et al., 2008, S. 10, eigene Übersetzung). Rowley u. Slack sehen ein Literature Review zudem als eine Zusammenfassung eines Themengebietes, welches eine Unterstützung bei spezifisch gestellten Forschungsfragen gibt (Rowley u. Slack, 2004, S. 31). Sie fügen hinzu, dass jeder Forscher im Bilde der existierenden Literatur in seinem Feld sein muss (Rowley u. Slack, 2004, S. 31) und Wissen

„[...] about a particular field of study, including vocabulary, theories, key variables and phenomena, and its methods and history“

besitzen sollte (Randolph, 2009, S. 2). Wissen über ein Forschungsfeld zu erlangen, sehen auch Schryen et al. als eine notwendige und charakterisierende Eigenschaft eines Literature Reviews an und formulieren:

„A literature review provides both a comprehensive synthesis and an interpretation of the body of knowledge of a specified domain“ (Schryen et al., 2015, S. 3).

Entsprechend muss im Zuge dieser Forschungsarbeit ein (systematisches)<sup>1</sup> Literature Review durchgeführt werden. Dass dieses nicht immer stattfindet,

<sup>1</sup> Kitchenham u. Charters unterscheiden ein Literature Review von einem systematischen Literature Review (auch systematischen Review genannt), obwohl in der Literatur die Begriffe häufig synonymisch verwendet werden und dabei eine Systematik implizit unterstellt wird (Randolph, 2009; Okoli u. Schabram, 2010). Bei einem systematischen Review wird die Nachvollziehbarkeit der Methode anhand einer ausführlichen Dokumentation in den Vordergrund gestellt. Sie nennen eine Reihe von Eigenschaften von systematischen Literature Reviews in (Kitchenham u. Charters, 2007, S. 4f.) und fügen an, dass es sich dabei um eine Sekundärstudie handelt - eine Bezeichnung die auch so von Cooper konnotiert wird (Cooper, 1988; Cooper, 1998). Kitchenham u. Charters definieren eine Sekundärstudie als: „A study that reviews all the primary studies relating to a specific research question with the aim of integrating/synthesizing evidence related to a specific research question“ (Kitchenham u. Charters, 2007, S. vi).

bzw. die Qualität eines solchen nicht in allen Fällen zufriedenstellend ist, wird von Randolph kritisiert (Randolph, 2009, S. 1)<sup>2</sup>. Gründe hierfür sehen Boote u. Beile in einer fehlenden Systematik bei der Anfertigung eines Reviews, was sie darauf zurückführen, dass die Analyse und Synthese von Forschungsliteratur keinen oder nur einen geringen Teil der Hochschulausbildung ausmachen (Boote u. Beile, 2005, S. 5). Entsprechend der durch Randolph formulierten Kritik, nennt dieser Gründe, welche für die Anfertigung eines Literature Reviews unter der Zuhilfenahme eines strukturierten Vorgehens sprechen. Für ihn steht nicht allein der Nachweis im Vordergrund, ein fundiertes Wissen in einem Themenbereich zu haben, vielmehr sieht er einen Publikationszweck, um die Forschung zu bereichern sowie die Möglichkeit eine Forschungscommunity zu einem Thema kennen zu lernen und zu unterstützen (Randolph, 2009, S. 2)<sup>3</sup>. Durch die Vorstrukturierung der einzelnen Analyseschritte, sowie den Einsatz von etablierten Vorgehensmodellen bei deren Durchführung, soll dieser mangelnden Strukturierung vorgebeugt werden.

Zahlreiche weitere Ziele eines Literature Reviews (neben denen, die bereits in den Begriffsdefinitionen erkennbar sind), können allgemein in der Literatur identifiziert werden. Basierend auf den Beiträgen von Boote u. Beile (2005, S. 3 ff.), Fink (2014, S. 3 ff.), Randolph (2009, S. 2 f.), Okoli u. Schabram (2010, S. 2 f.), Rowley u. Slack (2004, S. 32), Schryen et al. (2015, S. 8f.) sollen für das Literature Review in dieser Forschungsarbeit die folgenden Ziele gelten.

- Das Literature Review soll eine (strukturelle) Perspektive in Bezug auf wichtige Game Design-Elemente (Domänenkonzepte) in spielifizierte Lehr-Lernarrangements einnehmen,
- die innere Struktur von der identifizierten Domänenkonzepte aus spielifizierten Lehr-Lernarrangements herausarbeiten,
- Beziehungen und Theorien zwischen Domänenkonzepten und Anwendungspraktiken identifizieren und exzerpieren,
- Variationen bei Domänenkonzepten identifizieren und
- als Teil einer Forschungsagenda das Ergebnis des Reviews für einen weiteren Analyseschritt als Input bereitstellen, um die neuen Forschungsergebnisse mit einer anschließenden Lernplattformanalyse in Beziehung zu setzen und darauf aufzubauen.

<sup>2</sup> Eine Übersicht über Kritiker von bestehenden Literature Reviews sowie weitere Unterstützer dieser These, sind in (Randolph, 2009, S. 1) zu finden.

<sup>3</sup> Siehe hierzu auch Fink (2014, S. 9 ff. ) oder Okoli u. Schabram (2010, S. 2 ff.)

Für die Erreichung der in der Literatur formulierten Ziele für dieses Literature Reviews wird auf eine Methodik zur Anfertigung eines solchen zurückgegriffen. Eine bewährte und weit verbreitete Methodik zur systematischen Anfertigung eines Literature Reviews stammt von Cooper und wurde 1988 erstmals publiziert (Randolph, 2009, S. 2; Cooper, 1998; Cooper, 1988). Hierbei handelt es sich um eine quantitative Methodik zur Synthese von primären Forschungsstudien (Cooper, 1998, S. 4). Sie konzeptualisiert ein mögliches Vorgehen, wenngleich die Anordnung und Ausprägung der einzelnen Schritte zur Anfertigung eines Reviews Variationen zulässt (Cooper, 1998, S. 5), um beispielsweise bestimmte Schritte im Vorgehen durch qualitative Ansätze zu substituieren.

#### 4.2.1 *Verwandte Untersuchungen*

Die Spielifizierung der Lehre findet immer mehr Anklang und Verbreitung (Caponetto et al., 2014, S. 50; Hamari et al., 2014). Zum Zeitpunkt der Anfertigung dieses Literature Reviews, existieren Reviews, welche im Folgenden exemplarisch vorgestellt werden. Daraufhin wird begründet, warum diese für das Forschungsvorhaben zwar wertvolle Ideen liefern können, jedoch nicht als alleinige Basis ausreichend sind.

Hamari et al. beschäftigen sich innerhalb eines Literature Reviews mit empirischen Studien zu Gamification (Hamari et al., 2014). Sie untersuchen dabei, was die empirischen Studien zu Gamification bisher fokussierten in puncto motivationales Angebot, psychologische und verhaltensbeeinflussende Ergebnisse. Das Review hatte das Ziel eine Übersichtsarbeit zu schaffen, um die Frage zu beantworten: „Does Gamification work?“ (Hamari et al., 2014, S. 3029). Gestützt wurde das Literature Review auf eine Onlinesuche von einschlägigen Datenbanken (z.B. ACM Digital Library, Scopus etc.), bei der in Kombination mit einer Tiefensuche insgesamt 24 empirische Studien (davon zwei qualitativ, siebzehn quantitativ und fünf mixed-methods) für die Auswertung herangezogen werden konnten. In der Kategorie des motivationalen Angebots identifizierten die Autoren Points, Leaderboards und Achievements als die am häufigsten vorkommenden Mechanics. Außerdem fanden sie die folgenden Gamification Elemente: Levels, Story/Themes, Clear Goals, Feedback, Rewards, Progress, Challenge. Hinzukommend berichten sie über verhaltensbeeinflussende Ergebnisse durch den Einsatz von Gamification, welche durch Studien (Experimente oder statistische Analysen) mit existierenden Gamification-Lösungen erstellt wurden. Die psychologischen Ergebnis-

se der untersuchten Studien konzentrierten sich auf die Bereiche Motivation, Einstellung und Vergnügen bei an den Interventionen teilnehmenden Personen. Bei den untersuchten Studien wurde ebenso der Kontext erfasst. Er bezieht sich auf den „[...] core service or an activity being gamified“ (Hamari et al., 2014, S. 3028). Solche Studien, die in einem Lehr-Lernkontext erhoben wurden, zeigten positive Ergebnisse und damit eine Steigerung bezogen auf die Motivation, die Nachhaltigkeit bei der Teilnahme an Lehr-Lernaktivitäten und dabei ein erhöhtes Maß an Vergnügen während der Durchführung. Ebenso wurde von Begleiterscheinungen berichtet, wie z.B. erhöhter Wettbewerb, welche von den Teilnehmenden als vereinzelt negativ erachtet wurden. Zusammenfassend beantworten die Autoren dieses Literature Reviews die eingangs gestellte Forschungsfrage positiv, jedoch merken sie an, dass einige Einschränkungen bestehen können, die es zu beachten gilt. Beispielsweise sei der Überraschungseffekt (Neuheitseffekt) bei wiederholtem Einsatz von spielifizierten Aktivitäten nicht mehr gegeben, was zu einer Monotonie bei den Teilnehmenden und schließlich zu einer Umkehrung der gewünschten Effekte (Motivation, Nachhaltigkeit, Vergnügen) führen kann (Hamari et al., 2014; Kapp et al., 2014). Weiterhin geben die Autoren zu bedenken, dass die Studien aufgrund einer gemeinsamen wissenschaftlichen Basis nicht immer vergleichbar sind, was die Erstellung einer Metastudie nicht zulassen würde. Sie fordern damit Methodologien<sup>4</sup> ein, mit denen Gamification in einen wissenschaftlich vergleichbaren Kontext gestellt werden kann und dabei die Qualitäten der Teilnehmenden berücksichtigt (Hamari et al., 2014, S. 3029).

de Sousa Borges et al. führten eine systematische Mapping Studie nach Petersen et al. (2008) durch, mit dem Ziel die folgenden Forschungsfragen zu beantworten (de Sousa Borges et al., 2014, S. 217, eigene Übersetzung):

1. In welchen Lehrkontexten und Bildungsleveln wurde Gamification bisher am meisten untersucht?
2. Welche Typen von Studien kamen in Gamification und Lehre zum Einsatz?
3. Welche Gamification-Ansätze wurden am häufigsten im Kontext von Computer-Supported Collaborative Learning (CSCL) untersucht?

Für die Suche wurden, wie bei Hamari et al., einschlägige Datenbanken verwendet. Darin konnten insgesamt 26 relevante Werke identifiziert werden,

<sup>4</sup> Beispielhaft nennen sie die Möglichkeit Gamification Design-Elemente in ein System zu integrieren, diese über einen gewissen Zeitraum konstant zu halten und währenddessen Erhebungen durchzuführen, welche versuchen die Umgebungsparameter genauer zu erfassen (Hamari et al., 2014, S. 3030).

welche ohne eine Erweiterung um eine Tiefensuche, in die Auswertung eingeflossen sind. Für die Auswertung bildeten die Autoren auf Basis der Abstracts ein Kategoriensystem für die Einordnung der Materialien, welches aus den in Tabelle 6 dargestellten Kategorien bestand.

<b>Kategorie</b>	<b>Definition</b> Enthält alle Studien, die angeben, dass ...
Mastering Skills	... Gamification ein Mittel sein kann, die Fähigkeit von Studierenden zu erhöhen, sich erfolgreich mit komplexen oder repetitiven Lernaktivitäten auseinanderzusetzen.
Challenging	... spielifizierte Systeme, welche herausfordernde Aktivitäten bereitstellen, zur Verbesserung des Lernens beitragen.
Guidelines	... es Richtlinien gibt, wie Gamification in die Lehre integriert werden kann.
Engagement	... es Ansätze oder Strategien gibt, die das Interesse von Lernenden für ein Themengebiet nachhaltig steigern.
Improving Learning	... es spielifizierte Lösungen gibt, die das Ergebnis des studentischen Lernprozesses maximieren, in dem sie die Art wie gelernt wird, neue Wege aufzeigen.
Behavioural Change	... spielifizierte Systeme ein Mittel sein können, um eine positive Verhaltensänderung bei den Studierenden herbeizuführen.
Socialization	... mit Hilfe von spielifizierten Systemen das Lernen unter besseren Bedingungen stattfinden kann.

Tabelle 6: Auswertungskriterien und Bedeutung nach de Sousa Borges et al. (2014, S. 218, eigene Übersetzung)

Die Autoren kommen zu dem Ergebnis, dass Gamification vor allem im Bereich der Hochschulbildung anklang findet und dort am häufigsten mit dem Zweck der Motivationsförderung eingesetzt wird (Forschungsfrage 1). Ebenso geben sie an, dass der überwiegende Teil an Gamification-Literatur als Konferenz- bzw. Journalbeitrag publiziert wird, davon jedoch keine Erfahrungsberichte von Lehrenden oder Validierungen von neuen Methoden für die Erstellung von spielifizierten Lehr-Lernkontexten vorhanden sind (de Sousa Borges et al., 2014, S. 220) (Forschungsfrage 2). Vielmehr beschränkt sich die vorhandene Literatur weitestgehend auf die Motivationsförderung mit Hilfe von Gamification. Für die Beantwortung der dritten Forschungs-

frage konnten in dem untersuchten Material keine Informationen gefunden werden, was auf eine Forschungslücke hindeuten kann.

Caponetto et al. führten ein qualitatives Literature Review durch, welches auf einer Datenbasis von insgesamt 119 wissenschaftlichen Publikationen aufsetzt, die in einschlägigen Datenbanken gefunden werden konnten (Caponetto et al., 2014). Dabei setzten sie auf das Verfahren zur Anfertigung eines Literature Reviews von Rickinson u. May (2009). Die Autoren untermauerten das Review mit den folgenden Forschungsfragen (Caponetto et al., 2014, S. 53 ff., eigene Übersetzung):

1. Welche Ideen und Konzepte sind in Bezug auf Gamification und Lernprozesse bekannt?
2. Wird Gamification in der Literatur auf einem theoretischen Level behandelt oder gibt es empirische Studien?
3. Fokussiert die Spielifizierung von Lernprozessen eine bestimmte Zielgruppe?
4. Existiert eine klare Abgrenzung zwischen den Konzepten von Gamification und von Game-based Learning (GBL)?
5. Für welche Lehrgebiete werden Techniken von Gamification hauptsächlich adaptiert?

Auch die Ergebnisse dieses Reviews intendieren, dass Gamification in Lehr-Lernkontexten hauptsächlich zur Motivationsförderung eingesetzt wird und damit eine enge Verknüpfung zu der Annahme hat, dass der Einsatz von Gamificationansätzen einen positiven Einfluss auf den Lernprozess haben kann (Forschungsfrage 1) (Caponetto et al., 2014, S. 53). Diese in der Literatur identifizierten Ideen und Konzepte haben Caponetto et al. in einer Word Cloud zusammengefasst (siehe Abbildung 22).

Weiterhin stellen die Autoren fest, dass das Verhältnis zwischen theoretischen Arbeiten und empirischen Studien zu Gamification in ihrer Stichprobe nahezu ausgeglichen ist (Forschungsfrage 2). Die Frage nach der Zielgruppe beantworten die Forscher mit einem primären Einsatz von Gamification in einem formalen Bildungskontext. Allerdings merken sie an, dass eine Vielzahl an Arbeiten sich auf Trainingssettings beziehen (Forschungsfrage 3). Bezogen auf die vierte Forschungsfrage identifizieren sie Arbeiten, welche Gamification als ein Synonym für GBL bezeichnen. Auch wenn der überwiegende





Abbildung 22: Word Cloud über Konzepte und Ideen zu Gamification in der Lehre (Caponetto et al., 2014, S. 53)

Teil des Materials diese Position nicht teilt, erklären sich die Autoren dies dadurch, dass GBL-Einheiten (darunter zählen auch Serious Games) ein Teil eines breiter angelegten spielifizierten Lernprozesses sein können (Caponetto et al., 2014, S. 55) und auf den unterschiedlichsten Gebieten zum Einsatz kommt (z.B. Gesundheitswesen, Informatik, Software Engineering etc. - Forschungsfrage 5) (Caponetto et al., 2014, S. 55). Zusammenfassend lässt sich festhalten, dass das Design und die Planung von Lernaktivitäten nach Meinung der Autoren stärker in das Bewusstsein gerückt werden muss (Caponetto et al., 2014). Dabei sollen innovative Konzepte aus dem Bereich von Gamification stärker einbezogen werden.

Nah et al. führten ein Literature Review durch, um aufzuzeigen, welche Game Design-Elemente in Bildungs- bzw. Lernkontexten eingesetzt werden und was die Wirkungszusammenhänge ihres Einsatzes sind Nah et al. (2014b). Dabei konnten Sie die folgenden acht Game Design-Elemente identifizieren, welche in der untersuchten Literatur am häufigsten auftraten: Points, Levels/Stages, Badges, Leaderboards, Prizes/Rewards, Progress bars, Storyline, Feedback (Nah et al., 2014b, S. 406). Diese werden von den Autoren in Bezug zueinander gesetzt und deren unterschiedlichen Ausprägungen und Ziele näher erläutert. So beschreiben sie beispielsweise das Game Design-Element Point als eine Form von Rewards, eine Art des Aufwands, um den Fortschritt hinsichtlich eines Ziels auszudrücken oder eine Möglichkeit den momentanen Status eines Spielers zu ordinieren (Nah et al., 2014b, S. 405). Points werden häufig in der untersuchten Literatur als motivationsfördernd und verbindlichkeitssteigernd charakterisiert (Nah et al., 2014b, S. 402 f.).

Ebenso beschreiben die Autoren beispielhaft Plattformen und Systeme (z.B. *DevHub*<sup>5</sup>, *Stackoverflow*<sup>6</sup>, *Microsoft Ribbon Hero*<sup>7</sup>), welche spielifizierte Ansätze verfolgen, um (lernende) Benutzer dazu zu motivieren sich mit (Lern-) Inhalten auseinander zu setzen. Wie auch Hamari et al. kritisieren sie, dass es wenige methodengestützte und damit vergleichbare empirische Studien zu einem Einsatz von Gamification und dessen Wirken in Lehr-Lernkontexten gibt (Nah et al., 2014b, S. 407). Sie schlagen vor, Forschungsvorgehen wie Design Science Research nach Hevner et al. (2004) für die systematische Entwicklung und anschließende Evaluation spielifizierter Systeme zu etablieren.

Surendeleg et al. untersuchten die Literatur in Form eines Literature Reviews nach den folgenden beiden Fragestellungen (Surendeleg et al., 2014, S. 1610, eigene Übersetzung):

1. Wie kann die Lehre interessanter gestaltet werden?
2. Welche Rolle spielt Gamification in den Lernprozessen von Erwachsenen?

Sie fokussierten dabei vor allem empirische Studien, theoretische Arbeiten sowie bisherige Literature Reviews zu einem Einsatz von Gamification in Lehr-Lernkontexten. Um sich ihrer ersten Forschungsfrage thematisch zu nähern, identifizieren Sie das Fogg'sche Modell<sup>8</sup> Fogg (2009) als ein grundlegendes Werkzeug, um eine positive und zielorientierte Verhaltensänderung bei den Lernenden zu erreichen. Um den Motivationsanteil des Modells zu steigern, schlagen die Autoren den Einsatz von Game Mechanics vor. Dabei gehen sie jedoch nicht auf die Art und Weise ein, wann und unter welchen Umständen Game Mechanics eingesetzt werden sollen. Vielmehr wird die allgemeine Aussage vertreten, dass der Einsatz mit einer Motivationsförderung bei den Lernenden einhergehen kann (Surendeleg et al., 2014, S. 1612). Die Rolle von Gamification in der Lehre (2. Forschungsfrage) identifizieren sie als weitgehend unerforscht und stimmen damit mit der bereits vorgestellten Studie von Hamari et al. (2014) überein. Weiterhin identifizieren sie eine Forschungslücke, wonach bisherige Studien den Zusammenhang von Game Dy-

5 <http://www.devhub.com>, abgerufen am: 07.01.2016

6 <http://www.stackoverflow.com>, abgerufen am: 07.01.2016

7 <http://www.ribbonhero.com>, abgerufen am: 07.01.2016

8 Das Modell von Fogg verdeutlicht die Auffassung, dass eine (gezielte) Verhaltensänderung bei Menschen einer Ausgewogenheit der Faktoren Motivation, Fähigkeit und Trigger bedarf (Fogg, 2009, S. 1). Fehlt einer dieser drei Faktoren oder ist nicht ausreichend ausbalanciert im Vergleich zu den anderen beiden Faktoren, so kann nach Fogg keine Verhaltensänderung eintreten (Fogg, 2009, S. 2).

namics und Lerninhalt keine Beachtung schenkten (Surendeleg et al., 2014, S. 1613).

Seaborn u. Fels führten eine Metaanalyse durch, bei der sie die folgenden drei Ziele verfolgten (Seaborn u. Fels, 2015, S. 2, eigene Übersetzung):

- Eine Untersuchung der theoretischen und konzeptuellen Arbeiten zu Gamification, um herauszufinden, ob es eine gemeinsame Basis für den nach Ansicht der Autoren strittigen Terminus Gamification gibt.
- Die Bereitstellung einer vielfältigen Sichtweise aktueller Forschungsliteratur zu Gamification in unterschiedlichen Bereichen (z.B. Lehre oder soziale Netzwerke).
- Herausfinden, welche Verbindungen (sofern existent) zwischen den bisherigen Theorien und Anwendungen des Gamification Konzepts bestehen.

In ihrer Analyse untersuchten sie 31 Veröffentlichungen, die sie aus einer Treffergesamtheit von über 700 Materialien, gefunden in einschlägigen Online-Datenbanken, kriterienbasiert als relevant identifizierten. Dabei stellen sie fest, dass die von ihnen formulierte Hypothese, dass es keine gemeinsame Basis im Begriffsverständnis von Gamification gibt, zutreffend ist. Sie schlagen jedoch eine gemeinsame Basis in ihrer Definition von Gamification vor und definieren Gamification als „[...] the intentional use of game elements for a gameful experience of non-game tasks and contexts (Seaborn u. Fels, 2015, S. 8). Außerdem identifizieren und synthetisieren sie Gamification-spezifische konzeptuelle Frameworks, welche noch nicht praktisch angewendet wurden und damit der Beweis für deren Wirksamkeit nicht erbracht werden kann (Seaborn u. Fels, 2015, S. 13). Aus der Aussage der Autorinnen kann damit eine Forschungslücke abgeleitet werden, auf Basis derer ein standardisiertes Framework für das Design von Gamification in den unterschiedlichen Kontexten erforderlich ist, wonach strukturierte Evaluationen erfolgen können. Dies begründet sich weiterhin durch die Tatsache, dass in der untersuchten Literatur festgestellt werden konnte, dass bei der großen Mehrheit Gamification zur Anwendung kam, jedoch keine Aussagen darüber existieren unter welchen wissenschaftlich haltbaren Bedingungen eine solche Anwendung stattfand (Seaborn u. Fels, 2015, S. 29, 32). Für den Bereich Lehre, der Bereich in dem Gamification am häufigsten zum Einsatz kommt, stellen die Autorinnen im untersuchten Material fest, dass der Einsatz von Gamification als überwiegend positiv von den Lernenden bewertet wird (Seaborn u. Fels,

2015, S. 23, 31). Zu beobachten ist, dass es sich bei der untersuchten Literatur durchweg um Anwendungen von spielifizierten Systemen handelt, die entweder verfügbar waren oder eigens für eine spezifische thematische Nische entwickelt wurden. Über den Zeitraum der Nutzung derartiger Systeme treffen die Autorinnen keine Aussagen. Jedoch identifizieren sie die folgenden Gamification Elemente, die auf Basis ihrer Häufigkeit des Auftretens geordnet sind und einen Querschnitt durch alle untersuchten Bereiche darstellen (Seaborn u. Fels, 2015, S. 29): Points (18), Badges (15), Rewards (11), Leaderboards (11), Challenges (6), Status (5), Progression (3), Achievements (3), Avatars (2), Mini-Games (2), Roles (2), Narrative (1), Time Pressure (1) und Feedback (1). Abschließend ist festzuhalten, dass die Autorinnen sich für ein strukturiertes Vorgehen zum Design von spielifizierten Systemen aussprechen, welches unter Umständen eine optimale Kombination von Gamification Elementen berücksichtigt, welche immer funktionieren und auf die jeweilige Zielgruppe abgestimmt sein können (Seaborn u. Fels, 2015, S. 32).

Schlagenhauser u. Amberg schlagen in dem von ihnen durchgeführten Literature Review ein Klassifizierungsschema (siehe Abbildung 23) vor, welches für Gamification im Bereich der Informationssysteme geeignet ist (Schlagenhauser u. Amberg (2015)). Sie entwickelten es anhand von 34 Artikeln, welche allesamt in einschlägigen Journals oder Konferenzbänden zu finden sind und kriterienbasiert selektiert wurden. Für die Auswertung nutzten sie die Methodologie der Grounded Theory nach Glaser u. Strauss (1967).

Category	Subcategory
Application Area	Computer Aided Design, Collaborative Work System, Education, Health, Manufacturing & Production, Marketing
Conceptualizing Gamification	Definition, Design, Development Methods and Tools, Types
Management Issues	Evaluation, Integration, Planning Objectives, User Behavior, User Requirement, User Types
Theoretical Issues	Behavior, Motivation, Research

Abbildung 23: Klassifizierungsschema von Gamification-Literatur im Bereich der Informationssysteme nach Schlagenhauser u. Amberg (2015, S. 6)

Unter anderem wurde das identifizierte Material anhand des Kriteriums der Konzeptualisierung von Gamification untersucht. Dabei fokussierten die Autoren in ihrer Studie speziell die Bereiche (Subkategorien) Definition, Design, Entwicklungsmethoden und Tools sowie Anwendungstypen von Gamification (Schlagenhauser u. Amberg, 2015, S. 7). Sie kamen zu dem Ergebnis, dass Punkte, Badges und Leaderboards die am meisten eingesetzten Game Elemente darstellen, welche entweder in dem Kontext Lehre oder kollaborative Arbeitssysteme ihren Einsatz finden (Schlagenhauser u. Amberg, 2015, S. 9, 11).

Über das Design von Gamification und dafür geeignete Entwicklungsmethoden und Tools wird von den Autoren bis auf die als relevant identifizierten Game Elemente keine Aussage gemacht, obwohl dies zum Forschungsinteresse zählt. Sie stellen fest, dass in Bezug auf spielifizierte Informationssysteme „[...] still limited information about technology and how it should be applied“ (Schlagenhauser u. Amberg, 2015, S. 11) vorherrscht. Unterstützt wird diese These durch die vergleichsweise geringe Anzahl an Artikeln, welche sich überhaupt diesen Themen widmen. So werden Entwicklungsmethoden und Tools nur in einem Fünftel der untersuchten Materialien erwähnt bzw. näher darauf eingegangen (siehe z. B. Kappen u. Nacke (2013), Rojas et al. (2013) Knaving et al. (2013)).

Dicheva et al. führten eine Systematische Mapping Studie nach Petersen et al. (2008) durch und gingen dabei den Fragen nach, in welchen Lehr-Lernkontexten Gamification bisher eingesetzt wurde und welche Game Elemente für einen solchen Einsatz herangezogen wurden (Dicheva et al., 2015, S. 2). Die Auswahl des Untersuchungsmaterials erfolgte Kriterien-basiert. So wurde Literatur der letzten 5 Jahre in einschlägigen Datenbanken gesucht und schließlich 34 Materialien für die darauffolgende Auswertung identifiziert. Für die Auswertung wurden Variablen genutzt, anhand deren jedes Material durchsucht wurde: Game Elemente, Applikationstyp, Lehrlevel, akademisches Fach, Implementierung und berichtbare Evaluationsergebnisse (Dicheva et al., 2015, S. 3). Die Variable der Game Elemente unterteilen die Autoren in zwei weitere Subvariablen, die Game Mechanics (Deterding, 2013b) und „educational gamification design principles“ (Dicheva et al., 2015, S. 4). Als am Häufigsten in der Lehre eingesetzten Game Mechanics konnten die Autoren Points, Badges, Levels, Progress Bars, Leaderboards, virtuelle Währungen sowie Avatare identifizieren (Dicheva et al., 2015, S. 4). Bezogen auf die Subvariable *educational gamification design principles* geben sie an, dass „[...] visual status, social engagement, freedom of choice, freedom to fail and rapid feedback“ (Dicheva et al., 2015, S. 6) am Häufigsten zu finden waren. Der Einsatz von Gamification in der Lehre erfolgte in den meisten Fällen in Form eines Blended Learning Kurses in Hochschulen oder Weiterbildungsangeboten. Dabei widmeten sich die Bildungsangebote hauptsächlich den Themen der Informatik und nutzten hierfür eigens entwickelte Plattformen und Apps oder Plug-Ins für bestehende Learning Management Systeme (LMS), wie *Moodle*. Die Evaluationsergebnisse, sofern evaluiert<sup>9</sup> wurde, waren im überwiegenden Teil positiv. Das lässt nach Dicheva et al. den Schluss zu, dass der Einsatz von Ga-

<sup>9</sup> Die Autoren kritisieren, dass eine Vielzahl von Arbeiten, bei denen Gamification in Lehr-Lernkontexten eingesetzt wurde, keine Einordnung in einen wissenschaftlichen Kontext

mification durchaus einen positiven Effekt auf das Lernen haben kann, sofern dieser gut geplant und technologisch adäquat unterstützt sowie entsprechend korrekt angewandt wird (Dicheva et al., 2015, S. 10). Bezogen auf den technologische Unterstützung formulieren sie abschließend jedoch eine essentielle Einschränkung:

„The lack of proper technological support is one of the major general obstacles for applying game elements to education. Thus, the development of software tools that can efficiently support the gamification of various educational contexts would contribute to a larger scale adoption as well as research on the feasibility and efficacy of the gamification of education“ (Dicheva et al., 2015, S. 10).

Zusammenfassend ergibt sich bei den hier aufgeführten Literature Reviews ein Bild, welches eine Ableitung von potentiell relevanten und noch nicht oder nur marginal bearbeiteten Themen zulässt (siehe Tabelle 7). So wird in der vorhandenen Literatur oftmals kritisiert, dass ein Einsatz von Gamification in Lehr-Lernkontexten häufig unstrukturiert erfolgt, da beispielsweise keine technologische Unterstützung existiert, welche eine durchgehende und kontextsensitive Planung und anschließende Dokumentation erlauben würde (Schöbel et al., 2016, S. 2; de Marcos et al., 2016). Seaborn u. Fels formulieren die These, dass „[...] a lack of empirical research and standards of practice for design and implementation“ (Seaborn u. Fels, 2015, S. 2) ein Hindernis darstellen. Dieser These wird außerdem hinzugefügt, dass es einer solchen systematischen Unterstützung in Design und Technologie bedarf und dass dadurch neue Wege in der Forschung um das Thema Gamification der Lehre beschritten werden können (siehe z.B. Dicheva et al. (2015)). Nicht zuletzt wäre eine Vergleichbarkeit seitens der Realisierung von Gamificationansätzen in der Lehre gegeben. Damit könnte der Kontext eines Einsatzes in den Vordergrund gerückt werden und beispielsweise wie von Hamari et al. vorgeschlagen, der technologische Aspekt über mehrere Durchläufe konstant gehalten werden, um daraus in einem ersten Schritt eine Aussage über die Effektivität des Gamification-Ansatzes abzuleiten Hamari et al. (2014). Demnach wird deutlich, dass die vorgestellten empirischen Studien zum Thema Gamification in der Lehre sich nicht thematisch den Fragestellungen eines systematischen Designs und einer standardisierten Implementierung von Gamification-Anwendungen gewidmet haben. Um diesen Fragestellungen nachgehen zu können, soll ein Literature Review in Form einer Dokumentenanalyse durchgeführt werden, welche im Folgenden näher erläutert wird.

---

vornimmt sowie entsprechende Aussagen über den „Erfolg“ des Einsatzes in Form von Evaluationen fehlen (Dicheva et al., 2015, S. 10).

Quelle	Fokus	Design-Elemente	Design-vorgehensweisen	Design-prinzipien
Hamari et al. (2014)	Motivationales Angebot, psychologische und verhaltensbeeinflussende Ergebnisse	(x)	o	o
de Sousa Borges et al. (2014)	Erfassung verschiedener Arten von Studien	o	o	o
Caponetto et al. (2014)	Theoretische und empirische Konzepte	o	o	o
Nah et al. (2014a)	Game Design-Elemente und deren beschriebenes Wirken	x	o	o
Surendeleg et al. (2014)	Umgestaltung der Lehre und Rolle von Gamification	(x)	o	o
Seaborn u. Fels (2015)	Theorie und Einsatz von Gamification	x	o	o
Schlagenhaufer u. Amberg (2015)	Klassifizierung der Gamificationliteratur	x	o	o
Dicheva et al. (2015)	Einsatz von Gamification	x	o	x

Tabelle 7: Übersicht über vorgestellte Literature Reviews zum Thema Gamification in der Lehre (x = Nennung und Diskussion, (x) = Nennung, o = keine Nennung)

Nach dem Untersuchungszeitraum des Literature Reviews wurden noch weitere verwandte Untersuchungen mit vergleichbaren Fragestellungen durchgeführt. Diese sollen aus Gründen der Vollständigkeit Erwähnung finden: Pedreira et al. (2015), Schöbel et al. (2016), Lages Dos Santos (2016), Dichev u. Dicheva (2017).

#### 4.2.2 *Methodentheoretische Vorgangsbeschreibung*

Die zuvor aufgezeigten Literature Reviews konnten die zu Beginn dieses Kapitels formulierten Ziele nicht zufriedenstellend abdecken. Deshalb bedarf es eines Literature Reviews, welches spezifisch zur Erreichung der anfangs gestellten Zielformulierungen verhilft. Dabei soll sich etablierter Verfahren bedient werden, welche die durch DSR geforderte wissenschaftliche Stringenz erhöhen und damit positiv zur Qualität eines solchen Vorhabens beitragen. Die zu diesem Zweck eingesetzten Methoden sollen im Folgenden als Basis und zum Zwecke des Verständnisses und der Nachvollziehbarkeit aufgezeigt und dargelegt werden, wie diese im Zuge des Literature Reviews zum Einsatz kommen. Dabei soll sich im Wesentlichen auf die Arbeiten von Cooper (1988) und Mayring (2002) gestützt und diese kombiniert werden.

##### 4.2.2.1 *Problemformulierung und Zieldefinition*

In der Methodik nach Cooper werden fünf Schritte durchlaufen. Im ersten Schritt erfolgt die Formulierung bzw. Definition eines konkreten Problems oder Untersuchungsgegenstands. Dabei soll das Ziel des Reviews klar umrissen sowie „[...] the variables of interest, [...]“ (Cooper, 1998, S. 5) festgelegt werden. Ebenso werden diese zueinander in Beziehung gesetzt, um das Verständnis für die verschiedenen Konzepte und Theorien einer Domäne zu erzeugen. Nach Rowley u. Slack kann dies beispielsweise in Form einer Mind-map oder einer Konzeptmatrix abgebildet werden, welche sich im Zuge der Durchführung verändern dürfen (Rowley u. Slack, 2004, S. 36f.). Cooper gibt zu bedenken, dass die Definition der Variablen<sup>10</sup> unter dem Gesichtspunkt des Kontextes erfolgen muss, in dem sie angewendet werden, da diese sonst zu abstrakt für die Untersuchung sein können (Cooper, 1998, S. 13). Variablen sollen zudem operationalisiert definiert werden. Darunter versteht Cooper eine „[...] description of the observable events that determine if a concept is present in a particular situation“ (Cooper, 1998, S. 13). Dies bedeutet, dass die Bedingungen und Variablen, welche ein Konzept definieren, spezifiziert werden sollen, auch wenn diese während der Durchführung reevaluiert werden müssen (Elmes et al., 1995, S. 50):

„As the literature search proceeds, it is extremely important that synthesists take care to reevaluate the correspondence between the breadth or abstractness of their concepts and the variation in ope-

<sup>10</sup> Der Begriff „Variable“ lässt sich in der Literatur häufig als Alternative zu dem Begriff „Kategorie“ finden. „Variable“ ist eher mit einem quantitativen Forschungsstil konnotiert, während „Kategorie“ häufiger in qualitativen Ansätzen zu finden ist (Kuckartz, 2012, S. 45).



rations that primary researchers have used to define them“(Cooper, 1998, S. 15).

Als weiterer Bestandteil dieser Phase wird die in Tabelle 8 dargestellte Taxonomie zur Zieldefinition und Einordnung des Reviews herangezogen. Deren Zweck und Bestandteile werden im Folgenden näher beleuchtet.

Sowohl für die systematische Einordnung bestehender Literature Reviews, als auch als eine Orientierung für die Anfertigung neuer Literature Reviews schlägt Cooper eine Taxonomie vor (Cooper, 1988, S. 106). Diese begründet sich durch von ihm durchgeführte Studien, deren Ergebnisse als Primärressourcen für die Erstellung dieser Taxonomie herangezogen wurden (Cooper, 1988, S. 106). Cooper gibt an, dass sich sechs Charakteristika aus seinen Ergebnissen synthetisieren lassen, welche jeweils unterschiedliche Ausprägungen besitzen. Eine Übersicht stellt Tabelle 8 dar.

Name	Ausprägung
Fokus	Forschungsergebnisse Forschungsmethoden Theorien Anwendungen
Ziel	Integration Kritik Herausforderung
Perspektive	neutrale Darstellung Einnahme einer Position
Abdeckung	vollständig vollständig selektiv repräsentativ zentral
Organisation	historisch konzeptuell methodisch
Zielgruppe	Fachleute Wissenschaft Praktiker/politische Entscheidungsträger Öffentlichkeit

Tabelle 8: Taxonomie von Literature Reviews nach (Cooper, 1988, S. 109, eigene Übersetzung)

Der Fokus beschreibt das „Material welches von zentralem Interesse für den Reviewer ist“ (Cooper, 1988, S. 108, eigene Übersetzung). Dabei unterscheidet Cooper die Ausprägungen Forschungsergebnisse und -methoden, Theorien sowie Anwendungen (Cooper, 1988, S. 108). Er gibt außerdem an, dass die einzelnen Ausprägungen sich nicht gegenseitig ausschließen. Vielmehr treten diese häufig in Kombination auf (Cooper, 1988, S. 108).

Die zweite Eigenschaft ist nach Cooper das Ziel eines Reviews. Er nennt das Ziel einer Integration als eine Art der Ausprägung, welche sich wiederum in drei weitere Aspekte unterteilen lässt: Bei einer Generalisierung (1. Generalization (Cooper, 1988, S. 108)) werden durch das Zusammenfassen von Literatur in einem Forschungsfeld zentrale Aussagen zu diesem getroffen. Sind in der Literatur Widersprüche oder Konflikte zu finden, so lassen sich diese durch die Darstellung von neuen Konzepten auflösen (2. Conflict Resolution (Cooper, 1988, S. 108)). Eine weitere Möglichkeit der Integration sieht Cooper in der Entwicklung von sprachlichen Brücken (3. Linguistic Bridge-building (Cooper, 1988, S. 108)). Diese können notwendig sein, wenn Theorien oder Disziplinen über eine gemeinsame Sprache miteinander verbunden werden sollen. Damit kann sichergestellt werden, dass das Verständnis von Begriffen auf der gleichen Basis beruht. Die Kritik kann nach Cooper als eine weitere Ausprägung der Zielcharakteristik angesehen werden. Im Unterschied zu der Integration, bei der die verschiedenen Arbeiten miteinander verglichen werden, wird bei der Kritik von einem vorher definierten Kriterium ausgegangen, anhand dessen die verschiedenen Arbeiten bewertet werden. Es wird zudem eine Aussage darüber getroffen, ob eine Arbeit einem Kriterium standhält oder nicht. Bei der dritten Zielausprägung geht es um die Identifikation von zentralen Herausforderungen eines Forschungsgebiets. Diese lässt sich in weitere drei Aspekte unterteilen. Forschungsfragen können in den Fokus gerückt und analysiert werden, welche bereits durch frühere Arbeiten aufgegriffen wurden. Zudem können Forschungsfragen auf der Grundlage von bestehenden Arbeiten identifiziert werden, welche von Interesse für zukünftige Tätigkeiten in dem Forschungsgebiet sein können. Ebenso können methodische Probleme identifiziert werden, die dafür verantwortlich sind, dass sich ein Forschungsgebiet in der Vergangenheit nicht weiterentwickeln konnte. Cooper sieht bei der Eigenschaft Ziel ebenso eine Kombination der Ausprägungen als gewinnbringend an. Oftmals in der Literatur vorzufindende Formen bestehen aus der Kombination von Integration und Kritik oder Integration und dem Finden von zentralen Herausforderungen in einem Forschungsgebiet.

Als eine dritte Charakteristik, lassen sich Literature Reviews aus unterschiedlichen Perspektiven anfertigen. Laut Cooper beeinflusst die Perspektive eines Reviewers die Diskussion der in das Review einbezogenen Literatur. Er sieht eine grundsätzliche Unterscheidung in zwei Perspektiven: neutrale Einstellung und Einnahme einer Position. Bei einer neutralen Einstellung zu einem Thema, wird die einschlägige Literatur (zumindest anfangs) nicht interpretiert zusammengefasst und damit möglichst alle relevanten Sichtweisen erfasst. Dabei soll der Reviewer versuchen,

„[...] to distill the relevant works and to allocate attention to different theories, methods, issues, or outcomes in a manner that reflects their relative prominence in literature“ (Cooper, 1988, S. 110).

Wird hingegen durch den Reviewer eine Position eingenommen, gilt es, diese gegenüber anderen Argumentationen zu verteidigen und dessen Wert zu verdeutlichen. Cooper vergleicht diese Tätigkeit mit der eines Anwalts, der versucht die Beweise für seinen Mandanten hervorzuheben. Dabei können bestimmte Informationen ignoriert werden oder mit weniger Tiefe in das Review eingehen. Für ein Literature Review schließen sich die beiden Perspektiven nicht aus. Vielmehr betont Cooper, dass es mehr auf die Art und Weise ankommt, wie fair der Reviewer mit den unterschiedlichen Perspektiven aus den Werken umgeht und dabei seine eigene Position einbezieht:

„Thus, the perspective distinction relates more to how the works of others are treated than to the presence or absence of conclusions favoring one interpretation or another“ (Cooper, 1988, S. 110).

Er gesteht jedoch ein, dass es unwahrscheinlich ist, dass ein Reviewer eine vollständig neutrale Position vertritt.

Der Grad der Abdeckung stellt für Cooper eines der zentralen Unterscheidungskriterien von Reviews dar. Bei der ersten Ausprägung, der vollständigen Abdeckung, versucht der Reviewer die (annähernd) gesamte Literatur zu einem Thema zu erfassen, weniger detailreich zu beschreiben und daraufhin zu interpretieren. Bei einer vollständig selektiven Abdeckung, wird zwar ebenfalls die gesamte Literatur als Basis herangezogen, jedoch erfolgt eine Beschreibung von ausgewählten bzw. relevanten Werken in einer höheren Detailstufe. Auf Basis der ausgewählten Teilmenge können schließlich zusammenfassende Aussagen getroffen werden, was jedoch keine Garantie dafür ist, dass die Zusammenfassungen die gesamte Literaturbasis widerspiegeln. Bei dem dritten Grad der Abdeckung werden von dem Reviewer Beispielarbeiten anhand von definierten Charakteristiken ausgewählt, welche repräsentativ für weitere Arbeiten im selben Forschungsgebiet sind. Es wird demnach nicht die gesamte Literatur in einem Gebiet als Basis herangezogen. Die Charakte-

ristiken, welche zu der Auswahl bestimmter Werke führen, sollen diskutiert und reflektiert werden, damit eine Nachvollziehbarkeit gewährleistet werden kann. Bei der letzten Möglichkeit der Abdeckung werden von dem Reviewer zentrale Arbeiten (z.B. Standardwerke) herangezogen, welche ein Gebiet begründet oder maßgeblich zur (Weiter-) Entwicklung desselben beigetragen haben. Auch bei dieser Charakteristik ist eine Kombination von Ausprägungen möglich. Jedoch schließt sich eine Kombination von vollständig und vollständig selektiv für den selben Untersuchungsgegenstand aus. Die Ausprägungen repräsentativ und zentral hingegen, lassen sich sinnvoll zusammen einsetzen.

Für die Organisation der in ein Review einbezogenen Literatur, gibt es nach Cooper drei Möglichkeiten, die differenziert betrachtet werden. Bei einer historischen Anordnung, wird die Literatur chronologisch analysiert und beschrieben. Damit lassen sich beispielsweise die Ausbildungen von unterschiedlichen Forschungssträngen veranschaulichen und möglicherweise kausal begründen. Für eine konzeptuelle Anordnung gilt, dass Literatur, welche dieselben Ideen behandeln, zusammen beschrieben werden. Schließlich kann bei einer methodischen Organisation jene Literatur zusammen betrachtet werden, welche auf einer vergleichbaren methodischen Basis beruht. Hier sind ebenso Ausprägungskombinationen laut Cooper möglich. Er gibt beispielhaft an, dass es möglich sei, eine historische Anordnung heranzuziehen und innerhalb eines methodisch oder konzeptuellen Rahmens zu analysieren.

Die letzte Eigenschaft ist die Zielgruppe eines Reviews. Sie differiert in Fachleute, Personen aus der Wissenschaft, Praktiker oder öffentliche Entscheidungsträger, sowie die allgemeine Öffentlichkeit. Die Unterscheidung der Zielgruppe zeigt sich am deutlichsten in dem Schreibstil eines Reviewers. Während für Spezialisten im Gebiet der Wissenschaft mehr Fachwörter und Details im Schreibstil zu finden sind, gilt es für die allgemeine Öffentlichkeit diese möglichst grob-granular zu beschreiben und in der Essenz die Auswirkungen der Arbeiten zu erfassen. Durch diese Form des zielgruppenspezifischen Schreibens wird klar, dass eine Kombination der Zielgruppen nicht sinnvoll erscheint.

#### 4.2.2.2 *Suche nach einschlägiger Literatur*

Der zweite Schritt, die Datensammlung bzw. Suche von einschlägiger Literatur, beinhaltet die Entscheidung „[...] about the population of elements that will be the target of the study“ (Cooper, 1998, S. 8). Diese steht in einer direkten kausalen Abhängigkeit zu der entsprechenden definierten Ausgestaltung der vorherigen Phase des Literature Reviews.

#### 4.2.2.3 *Literaturtypen und deren Soll-Anteile am Review*

Rowley u. Slack sehen „articles in scholarly and research journals“ als den Kern eines Literature Reviews an (Rowley u. Slack, 2004, S. 32). Diese sind nach ihren Aussagen typischerweise peer-reviewed und von Forschenden geschrieben (Rowley u. Slack, 2004, S. 32; Cooper, 1998, S. 54, 74). Einen weiteren Vorteil sehen sie in der signifikanten Literaturbasis, welche vor allem in Journalartikeln zu finden ist, sogar dann, wenn das Thema eines Artikels nicht präzise im eigenen Suchfokus liegt (Rowley u. Slack, 2004, S. 32). Zudem werden Reviews durch nicht-akademische Literatur unterstützt. Die Autoren sprechen dieser Form jedoch einen geringeren Beitrag in der Fundierung und dem kritischen Umgang mit Modellen und Konzepten zu (Rowley u. Slack, 2004, S. 32). Der Zugang zu diesen Typen von Literatur ist vergleichsweise einfach über bibliographische Online-Datenbanken bzw. digitale Bibliotheken zu erlangen. Jedoch sind dort meist aus Lizenzgründen nicht alle Arbeiten einsehbar, was das Ergebnis einer Literatursuche einschränken kann. Dennoch spricht sich Cooper deutlich für die Nutzung einer computerbasierten Suche aus (im Gegensatz zu einer manuellen Suche), betont jedoch, dass ein „Browsing“ vor der eigentlichen Suche erfolgen soll, um sich einen entsprechenden Überblick zu verschaffen und gleichzeitig relevante Arbeiten zu identifizieren, welche später im Reviewergebnis auftauchen müssen (Cooper, 1998, S. 71). Dies trägt zur Qualitätssicherung der Suche nach Literatur bei.

Einen Überblick über Ideen und Entwicklungen in einem Forschungsbereich können Bücher geben. Sie bieten sich nicht nur wegen ihrer regelmäßigen inhaltlichen Überarbeitung für die Einbeziehung in ein Review an, vielmehr liefern sie wertvolle Literaturlisten, welche für eine spätere Tiefensuche von Relevanz sein können (Rowley u. Slack, 2004, S. 33).

Nach Cooper ist es vonnöten bei einer Suche möglichst mehrere Kanäle zu benutzen, um möglichst vielfältige Literatur zu erfassen. Er formuliert hierzu:

„As a rule, however, searchers must always employ multiple channels with different entry and access restrictions so that they minimize any systematic differences between included and unincluded studies“ (Cooper, 1998, S. 74).

#### 4.2.2.4 *Dokumentation des Suchprozesses*

Unabhängig von dem Forschungsgebiet, zu dem ein Review angefertigt wird, muss der Prozess der Literatursuche ausreichend dokumentiert werden. Randolph spricht hier von einem Grad der Nachvollziehbarkeit, welcher andere Forschende in die Lage versetzt, anhand der Dokumentation, und der Befol-

gung derselben Schritte mit denselben Bedingungen dieselbe Literatur zu finden (Randolph, 2009, S. 6). Er nennt dabei die folgenden Attribute, welche zu dokumentieren sind (Randolph, 2009, S. 7):

- Datum der Suche.
- Suchwörter und Suchwortkombinationen.
- Die Anzahl der Suchergebnisse zu jeder Suche.

#### 4.2.2.5 *Grad der Sättigung*

Die Suche nach Literatur kann abgeschlossen werden, wenn ein gewünschter Grad an Sättigung erreicht ist. Dies bedeutet, dass der Abschluss nicht an eine bestimmte Anzahl gefundener Werke gebunden ist. Vielmehr soll die Suche dann beendet werden, wenn bei einer Tiefensuche<sup>11</sup> keine neuen Ergebnisse mehr zu finden sind (Randolph, 2009, S. 7). Allerdings merkt Cooper hierzu an: „Generally speaking, however, focusing on only published studies is not advisable“ (Cooper, 1998, S. 75). Demnach ist es ratsam auch sogenannte Graue Literatur<sup>12</sup> mit in die Tiefensuche einzubeziehen.

#### 4.2.2.6 *Datenevaluation*

In der Phase der Datenevaluation geht es darum, zu entscheiden, ob bestimmte Literatur in ein Review einbezogen wird. Diese Entscheidung wird anhand von definierten Inklusions- bzw. Exklusionskriterien getroffen (Randolph, 2009, S. 7). Sie sind eine Hilfestellung dafür, dass eine eventuelle Voreingenommenheit des Reviewers – auch wenn dieser noch so objektiv zu sein scheint – sich möglichst wenig auf die Literaturbasis auswirkt (Cooper, 1998, S. 80f.). Beispiele hierfür können Kriterien wie die Sprache, inhaltlich-thematische Einschränkungen oder die Art der Literatur sein. Ist die Beurteilung anhand der Inklusions- bzw. Exklusionskriterien abgeschlossen, wird mit der Datenextraktion aus der gefundenen Literatur begonnen. Die gewonnenen Daten hängen von der Zielsetzung des Reviews ab (Randolph, 2009, S. 7) und es bedarf einer Dokumentation des Extraktionsprozesses.

---

<sup>11</sup> Die Tiefensuche ist eine mögliche Suchstrategie, welche zu einer gewünschten Sättigung führen kann. Sie beginnt bei der Durchsicht von Quellen eines Werks und daraufhin der Identifikation von weiterer möglicher Literatur und ist auch bekannt als Schneeballsuchverfahren. Eine Übersicht über weitere mögliche Suchstrategien, sowie deren Unterschiede ist in Rowley u. Slack (2004, S. 35f.) zu finden.

<sup>12</sup> Dabei handelt es sich um Literatur, welche nicht durch einen Verlag publiziert wurde, sondern von Autoren selbst publiziert wird. Beispiele hierfür sind Berichte oder Reporte.

#### 4.2.2.7 *Datenanalyse, Interpretation und Synthesetypen*

In der Phase der Datenanalyse und Interpretation werden die gewonnenen Daten aufbereitet und sinngebend miteinander kombiniert. Die Art und Weise wie diese Phase ausgestaltet ist, hängt wiederum maßgeblich von dem zu Beginn gesetzten Ziel des Reviews ab. Ebenso ist der Typ der Datenanalyse abhängig von den extrahierten Daten und kann unterschieden werden in eine quantitative, qualitative<sup>13</sup> oder mixed-method Synthese (Randolph, 2009, S. 8). Diesen drei Synthesetypen sind die unterschiedlichsten Methoden bzw. Vorgehensweisen zugeordnet. Allen gemeinsam ist jedoch die Orientierung an einem Kategoriensystem.

#### 4.2.2.8 *Qualitative Inhaltsanalyse nach Mayring*

Für diese Arbeit wird auf eine qualitative Synthese zurückgegriffen, unter Begründung der von Ritsert geäußerten Kritik an einer quantitativen Inhaltsanalyse. Er kritisiert, dass die quantitative Inhaltsanalyse vor allem vier Aspekte zu wenig berücksichtige: den Kontext von Textbestandteilen, latente Sinnstrukturen, markante Einzelfälle und das, was im Text nicht vorkommt Ritsert (1975). Zudem soll der Fokus des Literature Reviews, welches später in diesem Kapitel genauer beschrieben wird, nicht auf Häufigkeits-, Kontingenz- oder Identitätsanalysen liegen, sondern vor allem den oben genannten Schwächen quantitativer Verfahren entgegenwirken. Mayring formuliert den Grundgedanken einer qualitativen Inhaltsanalyse wie folgt:

„Qualitative Inhaltsanalyse will Texte systematisch analysieren, indem sie das Material schrittweise mit theoriegeleitetem am Material entwickelten Kategoriensystem bearbeitet“ (Mayring, 2002, S. 114).

Mit dieser Art von Verfahren (auch bekannt als qualitativ orientierte kategoriengeleitete Textanalyse (Mayring u. Fenzl, 2014, S. 544)) steht ein Werkzeug zur Verfügung, welches das technische Wissen der quantitativen Analyse zu Grunde legt und ebenso große Materialmengen bewältigen und Sinngehalte erfassen kann (Mayring u. Fenzl, 2014, S. 543; Lamnek, 2010, S. 460). Dabei fordert dieses Verfahren die exakte Definition von Auswertungsaspekten und -regeln, sodass ein „systematisches, intersubjektiv überprüfbares Durcharbeiten möglich wird“ (Mayring u. Fenzl, 2014, S. 545). Mayring unterscheidet dabei wiederum drei Grundformen der qualitativen Inhaltsanalyse: Zusammenfassung, Explikation und Strukturierung (Mayring, 2002, S. 115). Letz-

<sup>13</sup> Eine Übersicht über 17 unterschiedliche Verfahren zur qualitativen Analyse findet sich in Mayring (2002, S. 65ff.)

tere soll für die vorliegende Arbeit in der Phase der Datenanalyse herangezogen werden und verfolgt damit das Ziel

„[...] bestimmte Aspekte aus dem Material herauszufiltern, unter vorher festgelegten Ordnungskriterien einen Querschnitt durch das Material zu legen oder das Material auf Grund bestimmter Kriterien einzuschätzen“ (Mayring, 2002, S. 115).

#### 4.2.2.9 *Ablauf der strukturierenden qualitativen Inhaltsanalyse*

Im Folgenden soll der Ablauf einer strukturierenden qualitativen Inhaltsanalyse kurz erläutert werden. Zunächst werden die entsprechenden Analyseinheiten definiert. Darunter sind drei Typen zu verstehen (Mayring u. Fenzl, 2014, S. 546):

- **Kodiereinheit:** Definiert den minimalen Bestandteil aus einem Material, welcher für eine Auswertung herangezogen werden darf (semantische Einheit, Wort, Satz etc.).
- **Kontexteinheit:** Definiert welche Informationen für eine einzelne Kodierung herangezogen werden dürfen (Satz, Absatz, Interviewantwort, ganzes Interview, Zusatzkontextmaterial).
- **Auswertungseinheit:** Definiert die Materialportion, welcher ein Kategoriensystem gegenüber gestellt wird (ganzes Material, Materialteile, Mehrfachcodierungen etc.).

Im Anschluss daran erfolgt die Festlegung der Strukturierungsdimensionen. Sie wird theoriegeleitet begründet und aus den Forschungsfragen abgeleitet. Die einzelnen Dimensionen werden weiter verfeinert, indem sie in einzelne Ausprägungen aufgespalten werden. Die Summe aller Dimensionen und Ausprägungen bildet das Kategoriensystem. Dieses ist „sowohl Gegenstand entwicklungspsychologischer als auch erkenntnistheoretischer Überlegungen“ (Kuckartz, 2012, S. 41) mit dem Ziel, die Komplexität von Informationen systematisch zu reduzieren.

Bei der „[...] strukturierenden qualitativen Inhaltsanalyse handelt es sich um eine deduktive Kategorienanwendung“ (Mayring u. Fenzl, 2014, S. 548). Dies bedeutet, dass wie oben bereits angedeutet, das Kategoriensystem vorab und gestützt an der Theorie<sup>14</sup> entwickelt wird, um es anschließend auf die zu

---

<sup>14</sup> Beispielsweise abgeleitet an einer Forschungsfrage oder Bezugstheorie (Kuckartz, 2012, S. 69).



analysierenden Materialien anzuwenden (Kuckartz, 2012, S. 59f.). Eine Herausforderung dieses Ansatzes stellt die disjunkte und erschöpfende Formulierung der einzelnen Kategorien dar (Kuckartz, 2012, S. 61f.). Werden Kategorien induktiv entwickelt, kann es vorkommen, dass eine Zuordnung einer Fundstelle zu einer Kategorie nicht immer eindeutig bzw. möglich ist. Dies führt dazu, dass das Kategoriensystem einer Überarbeitung in Form von einer Präzisierung, einer Modifizierung oder einer Differenzierung bedarf, welche während des Prozesses ersichtlich wird. Damit wird also nicht ausgeschlossen, dass zunächst mit deduktiv definierten (Haupt-) Kategorien eine Fundstellenanalyse begonnen wird, um anschließend beispielsweise Subkategorien (und deren Definitionen) induktiv am Material zu entwickeln, welche in das Kategoriensystem aufgenommen werden (Kuckartz, 2012, S. 62)<sup>15</sup>.

Die Formulierung der einzelnen Kategorien, deren Definitionen, Ankerbeispielen sowie Kodierregeln zu den einzelnen Kategorien wird in einem Kodierleitfaden zusammengefasst. Anhand dessen kann systematisch und nachvollziehbar bestimmt werden, welche Analyseeinheit welcher Kategorie zugeordnet werden kann. Der Kodierleitfaden wird nach einer ersten Fertigstellung pilothaft auf einige Materialien angewandt und bei Bedarf entsprechend abgeändert. Bei einem darauffolgenden Hauptmaterialdurchlauf werden die entsprechenden Fundstellen gekennzeichnet und erfasst. Anschließend erfolgt in einer weiteren Iteration die Bearbeitung und Extraktion der Fundstellen. Abschließend erfolgt eine Ergebnisaufbereitung, die nach Mayring u. Fenzl (2014) und abhängig von der ursprünglichen Zieldefinition des Reviews, angefertigt wird. Sie unterscheiden dabei die folgenden Ausprägungen:

- Formale Strukturierung: „Will die innere Struktur des Materials nach bestimmten formalen Gesichtspunkten extrahieren und zusammenfassen“ (Mayring, 2015, S. 99).
- Inhaltliche Strukturierung: „Will Material zu bestimmten Themen, zu bestimmten Inhaltsbereichen extrahieren und zusammenfassen“ (Mayring, 2015, S. 99).
- Typisierende Strukturierung: „Will auf einer Typisierungsdimension einzelne markante Ausprägungen im Material finden und diese genauer beschreiben“ (Mayring, 2015, S. 99).

<sup>15</sup> Dieser Fall ist unter einer deduktiv-induktiven Kategorienbildung bekannt, einer Mischform der zwei Strategien (induktiv und deduktiv) zur Kategorienbildung (Kuckartz, 2012, S. 69; Mayring, 2015, S. 99). Nach Kuckartz stellt die inhaltlich strukturierende Inhaltsanalyse einen typischen Anwendungsfall für diese Form dar (Kuckartz, 2012, S. 69).

- Skalierende Strukturierung: „Will zu einzelnen Dimensionen Ausprägungen in Form von Skalenpunkten definieren und das Material daraufhin einschätzen“ (Mayring, 2015, S. 99).

Mayring u. Fenzl merken an, dass die vier Zielausprägungen der strukturierenden qualitativen Inhaltsanalyse in der Phase der Festlegung von Strukturierungsdimensionen und der Phase der Ergebnisaufbereitung differenziert werden. Für die Datenanalyse dieses Reviews soll eine inhaltliche Strukturierung herangezogen werden, da diese nicht nur zweckmäßig der Arbeit dienlich ist, sondern auch in den ursprünglichen Anfertigungsprozess von Cooper problemlos integriert werden kann. Daher wird im Folgenden nur diese Form fokussiert. Der grundlegende Ansatz dieses methodischen Vorgehens besteht also darin, die „Stärken der qualitativen Inhaltsanalyse beizubehalten und auf ihrem Hintergrund Verfahren systematischer qualitativ orientierter Textanalyse zu entwickeln“ (Mayring, 2015, S. 50).

Das Ziel einer „inhaltlichen Strukturierung ist es, bestimmte Themen, Inhalte, Aspekte aus dem Material herauszufiltern und zusammenzufassen“ (Mayring, 2015, S. 103). Nach Abschluss der Bearbeitung der zu analysierten Materialien mittels des Kategoriensystem, erfolgt eine Ergebniszusammenfassung, welche beginnend bei der feingranularsten Unterkategorie und sich hierarchisch im Kategoriensystem nach oben hin (Zusammenfassung pro Hauptkategorie) fortsetzt.

Der beschriebene Ablauf wird in Abbildung 24 grafisch dargestellt und dient als Prozess für die Phase der Datenanalyse und Interpretation.

#### 4.2.2.10 Ergebnispräsentation

Das Ergebnis dieser Phase ist die schriftliche Dokumentation der im Zuge des Literature Reviews gewonnenen Erkenntnisse. Die Sorgfalt, mit welcher diese abschließende Phase eines Reviews gehandhabt werden sollte, beschreibt Cooper als maßgeblich für den wissenschaftlichen Erkenntnisgewinn indem er formuliert:

„Without careful attention by the researcher to how the investigation is interpreted and described, all efforts to conduct a trustworthy and convincing scientific study leading up to this point are for naught“ (Cooper, 1998, S. 157).

Das Review ist somit abgeschlossen wenn die Ergebnisse<sup>16</sup> mit der entsprechenden Zielgruppe geteilt werden.

<sup>16</sup> Cooper schlägt für die Erstellung eines wissenschaftlichen Beitrags eine entsprechende Strukturierung des Dokuments vor (siehe hierzu Cooper (1998, S. 159ff.)).

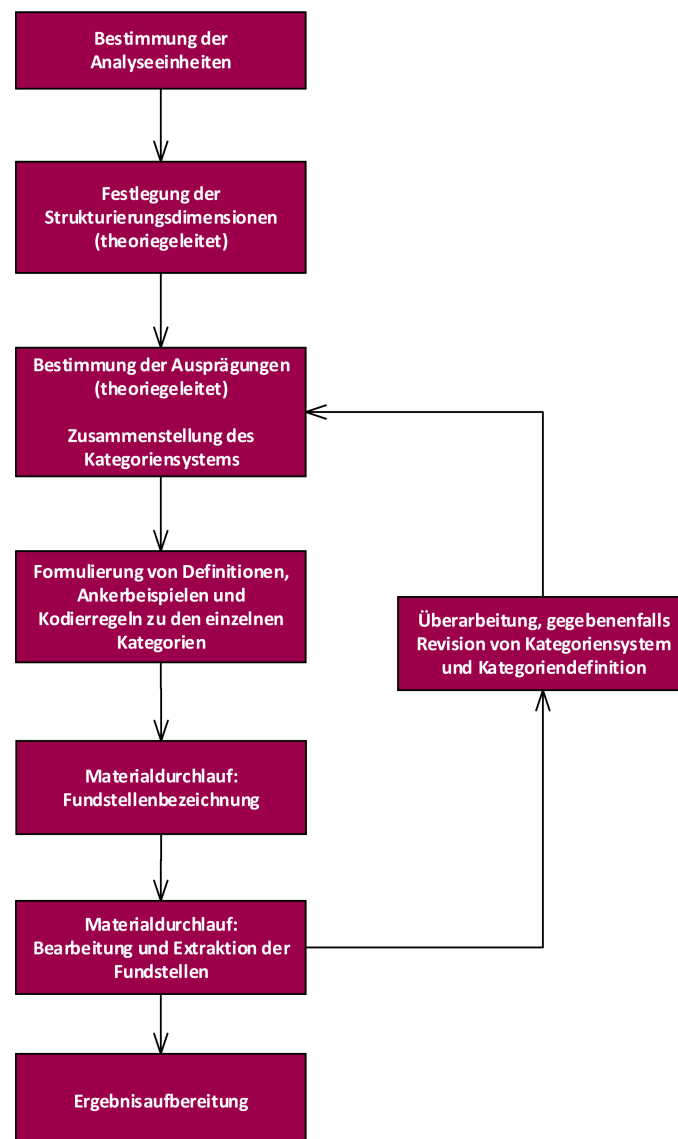


Abbildung 24: Ablaufmodell strukturierender qualitativer Inhaltsanalyse (Mayring, 2015, S. 98)

### 4.2.3 Instanziierung der beschriebenen Methodik

#### 4.2.3.1 Untersuchungsgegenstand und Forschungsfragen

Forschungsfragen können sich durch die unterschiedlichsten Gründe ergeben. Sie können sich beispielsweise durch bestehende Literatur ergeben, in der eine Forschungslücke erkannt wird oder in der Diskussion mit anderen Forschern (Morse, 1994, S. 220f.). Wie im vorangegangenen Kapitel aufgezeigt, Bedarf es einer zunehmender technologischen und methodischen Systematisierung, um den Einsatz von Gamification in Lehr-Lernaktivitäten unabhängig vom jeweiligen Kontext bewerten zu können (Nacke u. Deterding,

2017; Dichev u. Dicheva, 2017). Zudem wäre die Möglichkeit gegeben, einzelne Wirkungszusammenhänge von Game Design-Elementen und das Verhalten von Lernenden zu analysieren und Aussagen über Wirkungszusammenhänge zu treffen. Um dieses erhöhte Maß an Standardisierung zu erlangen, soll zunächst analysiert werden, welche Game-Design-Elemente vornehmlich in Lehr-Lernkontexten zum Einsatz kommen. Anschließend soll dieser Einsatzkontext näher beleuchtet werden, mit dem Ziel, Gemeinsamkeiten von unterschiedlichen Einsatzszenarien offen zu legen. Parallel dazu soll der Fokus auf der strukturellen Beschreibung von den eingesetzten Game-Design-Elementen liegen, vor allem bezogen auf deren Attribute, Zusammenhänge und Regeln denen diese beim Einsatz unterliegen. Konkreter bedeutet dies, dass es zunächst vonnöten ist, die innere Struktur von etablierten Game Design-Elementen zu verstehen, um herausfinden zu können, welche Möglichkeiten der Standardisierung beispielsweise durch den Einsatz von Abstraktion, bestehen. Die folgenden Forschungsfragen sollen in diesem Literature Review untersucht werden und greifen die zuvor identifizierten Arbeitsfelder auf:

**F.1.** Welche Elemente von Gamification werden in Lehr-Lernkontexten eingesetzt?

**F.2.** Wie können die identifizierten Elemente von Gamification strukturell beschrieben werden?

- **F.2.a.** Welche Attribute lassen sich identifizieren?
- **F.2.b.** Welche Zusammenhänge zu anderen GDEs und Objekten des Anwendungskontextes lassen sich feststellen?
- **F.2.c.** Welchen Regeln folgt der Einsatz von GDEs?

#### 4.2.3.2 *Festlegung der Analyseeinheiten*

Im Folgenden sollen die Analyseeinheiten festgelegt werden, um die Präzision der durchgeführten Analyse zu erhöhen (Mayring u. Fenzl, 2014, S. 553).

Für die Kodiereinheit (minimale semantische Einheit die ausgewertet werden kann) wird festgelegt, dass mehrere sinngebende Wörter (Mayring u. Fenzl sprechen hierbei von „bedeutungstragende Phrasen“ (Mayring u. Fenzl, 2014, S. 553)) mit Bezug zu den festgelegten Kategorien den kleinsten auszuwertenden Bestandteil bilden. Diese Festlegung scheint insofern sinnvoll, da bei der Datenextraktion der Kontext ebenso entscheidend ist, wie die codierte Einheit selbst. Eine Kodiereinheit<sup>17</sup> bestehend aus einem Wort wäre hier nicht

<sup>17</sup> Ein alternativer Terminus ist „Codierung“. Beide Begriffe bezeichnen die Zuordnung von Kategorien zu Textstellen bzw. zu Teilen des Untersuchungsmaterials (Kuckartz, 2012, S. 48).

ausreichend aussagekräftig (Kuckartz, 2012, S. 47). Bezogen auf die Kontexteinheit (legt fest welche Informationen für eine einzelne Kodierung herangezogen werden dürfen (Mayring u. Fenzl, 2014, S. 546)) geht diese mit der Festlegung der Kodiereinheit einher<sup>18</sup>. Für die Kontexteinheit gilt damit, dass diese mindestens dem semantischen Bereich der Kodiereinheit entsprechen muss. Sofern für die Interpretation oder das Verständnis einer Kodiereinheit der Kontext eine maßgebliche sinnstiftende Rolle spielt, kann es vonnöten sein, dass die Einheit größer gefasst ist als eine dazugehörige Kodiereinheit. Die Auswertungseinheit (Materialportion zur Gegenüberstellung eines Kategoriensystems) wird durch alle als relevant identifizierten Materialien dargestellt. Mehrfachzuordnungen von Bestandteilen eines Materials zu unterschiedlichen Kategorien sind zulässig, sofern diese unterschiedliche inhaltliche Aspekte, bezogen auf die Kategorien denen sie zugeordnet werden, darstellen. Mehrfachnennungen innerhalb eines Materials werden nicht erfasst, sofern sie sich nicht signifikant in dem Kontext in dem ein Materialbestandteil vorkommt, unterscheiden.

#### 4.2.3.3 Zielsetzungsformulierung

Für die Zielsetzung dieses Reviews wird auf Basis der Forschungsfragen zudem die bereits vorgestellte Taxonomie von Cooper (1988) herangezogen. Sie bietet einen strukturierten Überblick darüber, was mit diesem Literature Review durch Operationalisierung der Forschungsfragen erreicht werden soll und ist für dieses Review in Tabelle 9 dargestellt (die gewählten Ausprägungen sind fett markiert).

Im Fokus dieses Reviews stehen Forschungsergebnisse, Theorien sowie Anwendungen. Forschungsmethoden werden nur im Zusammenhang mit Forschungsergebnissen betrachtet, sofern eine für die Lehre relevante Kontextfassung der Methode aus der Literatur exzerpiert werden kann. Vor allem aus den Anwendungen und deren Evaluationen lassen sich wertvolle Erkenntnisse gewinnen, um die genannten Forschungsfragen beantworten zu können. Zudem können einschlägige Zusammenhänge im Sinne von Best-Practices bei einem Einsatz von Game Design-Elementen in Lehr-Lernkontexten identifiziert werden, welche sich möglicherweise zu einer Systematik abstrahieren lassen. Damit soll dieses Review das Ziel einer Integration verfolgen, welche in einem weiteren Schritt zu einer Generalisierung der identifizierten Konzepte führt. Für das Review wird eine neutrale Position eingenommen. Diese

<sup>18</sup> Kuckartz merkt an, dass die Kontexteinheit normalerweise nicht größer definiert wird, als die Analyseeinheit, mit der Ausnahme, dass Sonderfälle durchaus möglich, wenn auch nicht häufig sind (Kuckartz, 2012, S. 48).

Wahl der Perspektive begründet sich unter anderem dadurch, dass keine Meinung oder These verteidigt wird, vielmehr alle verfügbaren Ansätze gesammelt und integriert werden, womit die Forschungsfragen beantwortet werden können. Diese Sammlung erfolgt im Rahmen einer möglichst vollständigen aber selektiven Abdeckung durch die genannten Literaturquellen<sup>19</sup>. Eine andere Ausprägung der Abdeckung ist für das Ziel der vorliegenden integrativen Forschungsarbeit weniger geeignet, da möglicherweise wertvolle Ansätze nicht gefunden werden können und damit die Ergebnisvielfalt einschränken. Eine vollständige Abdeckung würde den Rahmen dieser Arbeit überschreiten und die vorgesehene Fokussierung zur Literatur gestützten Beantwortung der Forschungsfragen nur bedingt unterstützen. Bezogen auf die Organisation wird eine konzeptuelle Struktur gewählt. Eine historische Betrachtung spielt mit Sicherheit subtil bedingt durch die konzeptuelle Organisation eine Rolle, steht jedoch nicht im Vordergrund dieser Untersuchung. Das Gleiche gilt für eine methodische Organisation. Bei der Zielgruppe dieser Untersuchung handelt es sich in erster Linie um Fachleute bzw. Personen aus der Wissenschaft, wobei Randolph anmerkt, dass die Korrektoren und Reviewer einer Dissertation das Primärpublikum<sup>20</sup> darstellen (Randolph, 2009, S. 4).

#### 4.2.3.4 *Definition der Hauptauswertungskategorien*

Das Kategoriensystem ist ein essentielles Merkmal einer qualitativen Inhaltsanalyse. Begründet wird dies unter anderem dadurch, dass die Arbeit „mit einem Kategoriensystem einen entscheidenden Punkt [zur] Vergleichbarkeit der Ergebnisse“ (Abschätzung der Reliabilität) eines Reviews beiträgt (Mayring, 2015, S. 52). Über dessen Konstruktion sind in der Literatur jedoch nur vergleichsweise wenige Hilfestellungen zu finden (Mayring, 2015, S. 51; Krippendorff, 1980; Kuckartz, 2012, S. 41f., 59). Krippendorff bezeichnet die Definition der Kategorien als Kunst, was auf den kreativen Charakter dieses Definitionsprozesses hindeutet (Krippendorff, 1980, S. 76). Aufgrund der zweiten Forschungsfrage werden Hauptkategorien definiert, die eine Unterteilung der Ergebnisse anhand von spielifizierten Konzepten ermöglichen. Als Grundlage für diese Unterteilung dient das bereits vorgestellte Gamification Design-

19 Selbst die Unterstellung einer vollständigen Abdeckung wäre nur schwer umsetzbar, da alleine schon die Auswahl der Literaturdatenbanken eine Einschränkung bedeuten würde. Weitere Einschränkungen könnten aus Lizenzgründen, durch sprachliche Barrieren (andere Sprachen) oder begriffliche Unschärfe existieren, die den Anspruch der Vollständigkeit erheblich schmälern und damit eine Selektion implizieren.

20 Eine Ausweitung auf Praktiker und politische Entscheidungsträger kann nach Abschluss der Dissertation erfolgen. Praktiker können auch Lehrende sein. Der Begriff Praktiker unterliegt einem definitorischen Rahmen, welcher diejenigen einbezieht, welche sowohl Praktiker (Lehrende) als auch Wissenschaftler sind.

Name	Ausprägung
Fokus	<b>Forschungsergebnisse</b> Forschungsmethoden <b>Theorien</b> <b>Anwendungen</b>
Ziel	<b>Integration (Generalisierung)</b> Kritik Herausforderung
Perspektive	<b>neutrale Darstellung</b> Einnahme einer Position
Abdeckung	vollständig <b>vollständig selektiv</b> repräsentativ zentral
Organisation	historisch <b>konzeptuell</b> methodisch
Zielgruppe	<b>Fachleute</b> <b>Wissenschaft</b> Praktiker/politische Entscheidungsträger Öffentlichkeit

Tabelle 9: Instanziierung der Taxonomie von Literature Reviews nach (Cooper, 1988, S. 109, eigene Übersetzung)

Element Framework von Werbach (2014). Demnach sind die Hauptkategorien zunächst orientiert an Dynamics, Mechanics und Components sowie den jeweiligen Attributen, Zusammenhängen und Regeln mit anderen Game Design-Elementen als Subkategorie.

#### 4.2.3.5 Auswahl des Untersuchungszeitraumes

Der Zeitraum, zu dem die Daten erhoben werden, ist auf den 01.01.2009 bis 31.12.2014 festgelegt. Eine vor der eigentlichen Literatursuche durchgeführte Pilotisierung der für das Review genutzten elektronischen Datenbanken ergab, dass vor dem Jahr 2009 keine einschlägigen Ergebnisse zu dem Schlagwort Gamification (im deutsch und englischsprachigen Raum) gefunden werden können. Deterding et al. geben an, dass der Terminus Gamification vor der zweiten Hälfte des Jahres 2010 keine weite Verbreitung fand, obwohl die ers-

te dokumentierte und mit heutiger Definition vergleichbare Nutzung auf das Jahr 2008 datiert wird (Deterding et al., 2011a, S. 9). Laut Werbach und Hunter wurde der Begriff 2003 von Nick Pelling, einem britischen Programmierer, erstmal veröffentlicht, als dieser eine Beratungsfirma für spielerische Benutzerschnittstellen für elektronische Geräte ins Leben rief (Werbach u. Hunter, 2012, S. 25). Auch sie sprechen dem Begriff erst seit Mitte 2010 eine weite Verbreitung im heutigen Sinne zu (Werbach u. Hunter, 2012, S. 25f.).

#### 4.2.3.6 *Festlegung des Sprachradius*

Da Gamificationansätze in einem internationalen Diskurs stehen, ist es für die vorliegende Arbeit von Bedeutung, dass nicht ausschließlich regional begrenzte (deutschsprachige Literatur) berücksichtigt wird, sondern darüber hinaus ebenso internationale Werke in englischer Sprache mit in die Untersuchung einfließen.

#### 4.2.3.7 *Durchführung der Literatursuche*

Bei dem Suchprozess handelte es sich um eine computerbasierte Suche, welche im ersten Quartal 2015 durchgeführt wurde. Dabei wurden die in Tabelle 10 aufgeführten elektronischen Datenbanken durchsucht. Diese wurden ausgewählt, da sie sich durch eine hohe Publikationsanzahl zu dem Themengebiet auszeichnen und für bereits durch andere Forschende erstellte Literature Reviews im Forschungsgebiet Gamification erfolgreich für die Literatursuche herangezogen wurden (z.B. Hamari et al. (2014), Dicheva et al. (2015), Caponetto et al. (2014), Surendeleg et al. (2014)). Ebenso sind die unterschiedlichsten Literaturtypen in den ausgewählten Datenbanken vertreten. Hauptsächlich werden jedoch wissenschaftliche Artikel in den Datenbanken angeboten, welche den Hauptanteil in einem Literature Review stellen sollten (Rowley u. Slack, 2004, S. 32; Cooper, 1998, S. 74).

Datenbank/Quelle Akronym Link			Review
AIS Electronic Library	AISeL	<a href="http://aisel.aisnet.org/do/search">http://aisel.aisnet.org/do/search</a>	Hamari u. Koivisto (2014)
weiter auf nächster Seite			



Datenbank/Quelle	Akronym	Link	Review
Association for Computing Machinery Digital Library	ACM	<a href="http://dl.acm.org/dl.cfm">http://dl.acm.org/dl.cfm</a>	Hamari u. Koivisto (2014), Dicheva et al. (2014), Surendeleg et al. (2014)
Elton B. Stephens Company Host	EBSCO	<a href="http://search.ebscohost.com">http://search.ebscohost.com</a>	Hamari u. Koivisto (2014)
Education Resources Information Center	ERIC	<a href="http://eric.ed.gov">http://eric.ed.gov</a>	Dicheva et al. (2014)
Google Scholar	GS	<a href="http://scholar.google.de">http://scholar.google.de</a>	Hamari u. Koivisto (2014), Dicheva et al. (2014), Caponetto et al. (2014), Surendeleg et al. (2014),
Institute of Electrical and Electronics Engineers Explore	IEEE	<a href="http://ieeexplore.ieee.org/Xplore/home.jsp">http://ieeexplore.ieee.org/Xplore/home.jsp</a>	Dicheva et al. (2014), Surendeleg et al. (2014)
Elsevier Scopus	SC	<a href="http://www.scopus.com/">http://www.scopus.com/</a>	Dicheva et al. (2015)
Science Direct	SD	<a href="http://www.sciencedirect.com">http://www.sciencedirect.com</a>	Hamari u. Koivisto (2014), Dicheva et al. (2014)
Springer Link	SL	<a href="http://link.springer.com/">http://link.springer.com/</a>	Dicheva et al. (2014)
Web of Science	WS	<a href="http://webofknowledge.com">http://webofknowledge.com</a>	Hamari et al. (2014)

Tabelle 10: Ausgewählte elektronische Datenbanken für die Literatursuche

Sofern vorhanden, wurde für die einzelnen Suchanfragen ein erweiterter Suchmodus genutzt, um die vorgefertigten Queries nutzen zu können. Zudem war es vonnöten die Queries teilweise auf die Abfragesprache der jeweiligen Datenbank zu adaptieren. Dies geschah so, dass die Querysyntax angepasst, jedoch dabei keine Veränderung an der Semantik vorgenommen wurde. Somit kann davon ausgegangen werden, dass trotz dieser notwendigen Anpassungen die Treffermengen der einzelnen Datenbanken auf der gleichen semantischen Abfrage beruhen und damit eine gemeinsame Betrachtung in der späteren Analyse ohne Einschränkungen durchgeführt werden kann. Die Tabelle 11 dokumentiert die jeweiligen Suchstrings. Bei der Konstruktion der Queries wurde den Empfehlungen von Brereton et al. (2007) gefolgt, wonach die Suchterme aus der/den Forschungsfrage(n) abgeleitet, Alternativen und Synonyme daraus gebildet, diese in bereits bekannter Literatur als Keywords gegen geprüft und schließlich mit boolschen Operatoren verknüpft werden (Brereton et al., 2007).

Nr.	Querystring
Q1	„gamification“ and („education“ or „learning“ or „training“ or „teaching“ or „instruction“)
Q2	„gamified“ and („education“ or „learning“ or „training“ or „teaching“ or „instruction“)
Q3	„gameful“ and („education“ or „learning“ or „training“ or „teaching“ or „instruction“)
Q4	„gamify“ and („education“ or „learning“ or „training“ or „teaching“ or „instruction“)

Tabelle 11: Übersicht über Querystrings

Vor der eigentlichen Literatursuche wurde demnach vorab eine Pilotierung der Suchstrings durchgeführt, um deren Validität zu überprüfen und vorab relevante Materialien zu identifizieren. Vor allem Letztere können das Ergebnis der Suche qualitätssichern, in dem zu findende Werke bereits vor Beginn der eigentlichen Literatursuche bekannt sind (Cooper, 1998, S. 71). Insgesamt wurden in diesem Pretest aller Datenbanken 13 Werke identifiziert, welche sich durch ihre Zitationshäufigkeit<sup>21</sup> und damit wissenschaftliche Akzeptanz

<sup>21</sup> Zugrunde gelegt wurden als Metrik die kumulierten Zitationshäufigkeitsangaben der jeweiligen Datenbanken.

in der wissenschaftlichen Gemeinde auszeichnen, sowie Literatur, welche für die Forschungsfragen von einschlägiger Relevanz ist.

Zur jedem Treffer wurde die vollständige Zitation sowie das jeweilige Textmaterial (sofern aus Lizenzgründen vorhanden) erfasst. Ebenso wurde der Suchzeitraum auf der jeweiligen Datenbank dokumentiert, mit dem Ziel die Transparenz und Nachvollziehbarkeit des Suchprozesses zu erhöhen (Randolph, 2009, S. 7).

#### 4.2.3.8 Inklusions- und Exklusionskriterien

Das Material wurde anhand von Inklusionskriterien gefiltert. Folgende formale Inklusionskriterien führten dazu, dass das Material in die Analyse einbezogen wurde:

- Es handelt es sich (nach dem Begriffsverständnis der Cooper Taxonomie (Cooper, 1988)) um entweder ein wissenschaftliches Forschungsergebnis, eine entwickelte Theorie oder um eine dokumentierte sowie evaluierte Anwendung.
- Das Literaturformat ist entweder ein Buch (Monographie), ein Journal- oder Magazinbeitrag, ein Beitrag<sup>22</sup> in einem Tagungsband, Konferenzband oder Sammelwerk, eine Dissertation oder ein Report.
- Das Material ist in Englisch oder Deutsch verfasst.

Ebenso wurden Materialien aufgrund der folgenden inhaltlichen Exklusionskriterien von der Ergebnismenge ausgeschlossen:

- Gamification wird nur sekundär in dem Material behandelt (weniger als 60% der Länge des Materials). Der inhaltliche Fokus liegt beispielsweise auf allgemeiner Spieleentwicklung oder Serious Games.
- Gamification wird nicht in Bezug zu formalen<sup>23</sup> Lehr-Lernprozessen gesetzt.
- Publikationen, die nicht über die reine Beschreibung der Anwendung (im Sinne der Cooper Taxonomie (Cooper, 1988)) von Gamification hinausgehen, also ohne diese in einen wissenschaftlichen Kontext zu stellen (z. B. in Form von Evaluationsergebnissen oder wissenschaftliche Fundierung für das Design der Anwendung).

<sup>22</sup> Gemeint sind Short und Full Paper. Nicht eingeschlossen sind kurze Beiträge wie Abstracts, Workshopproposals etc.

<sup>23</sup> Nach dem Memorandum über Lebenslanges Lernen findet formales Lernen „in Bildungs- und Ausbildungseinrichtungen statt und führt zu anerkannten Abschlüssen und Qualifikationen“ (Europäische Kommission, 2000, S. 9).

- Publikationen einer oder mehrerer Autoren, die keinen neuen inhaltlichen Beitrag zum Erkenntnisgewinn bzgl. der gestellten Forschungsfragen leisten.

#### 4.2.3.9 Trefferübersicht und Auswertungsergebnisse

Die Tabelle 12 dokumentiert die Anzahl der Treffer, welche bei der Suche zunächst identifiziert werden konnten. insgesamt handelte es sich um  $n=3656$  Treffer. Nach Anwendung der Inklusions- bzw. Exklusionskriterien auf das bei der Literatursuche als potentiell relevante Material, ergibt sich dadurch die Treffermenge, bestehend aus  $n=322$  Arbeiten, welche potentiell für die Auswertung relevant ist. Aufgrund einer qualitativen Auswertung bedarf es Aufwands-bedingt einer weiteren Einschränkung der relevanten Treffermenge.

Die ausgewertete Treffermenge umfasste schließlich  $n=33$  Arbeiten. Begründet wird die Anzahl dadurch, dass aus den potentiellen Treffern von  $n=322$  mittels Convenience-Sampling (Flick, 2007, S. 166f.) zwei Auswahlverfahren kombiniert wurden, die *Ad-hoc-Stichprobe* sowie die *Quotenstichprobe*. Erstere sorgt dafür, dass sich die auszuwertende Ergebnismenge durch eine möglichst große Datenvielfalt auszeichnet (Streuungsprinzip (Akremi, 2014, S. 274)), jedoch Probleme in der Verallgemeinerung der Aussagen mit sich bringt, weshalb sie nicht als alleinige Stichprobenstrategie angewendet werden soll (Akremi, 2014, S. 273). Deshalb wurde diese mit einer Quotenstichprobe kombiniert, welche auf der Grundlage von vorherigen Merkmals- oder Dimensionsausprägungen, eine bestimmte Anzahl an Elementen zu finden abzielt (Akremi, 2014, S. 273; Flick, 2007, S. 167). Die folgenden Merkmalsausprägungen basierend auf den inhaltlichen Dokumenttypen und deren Soll-Anteile an der Ergebnismenge wurden somit definiert und damit der Empfehlung von Akremi (2014) gefolgt, diese aus Gründen der Transparenz darzustellen:

- **Typ 1 ( $n=11$ ):** Konzeptuelle Beschreibung eines spielifizierten Modells, einer Theorie oder Anwendung.
- **Typ 2 ( $n=11$ ):** Empirische Studie zu einer online oder offline spielifizierten Anwendung.
- **Typ 3 ( $n=11$ ):** Durch Theorie oder Empirie gestützte Best-Practices zum Einsatz von Game Design-Elementen.

Die inhaltliche Differenzierung der Typen ist an den Forschungsfragen orientiert. Die Zuordnung eines Untersuchungsgegenstands zu einem bestimmten Typ erfolgte anhand des Inhalts eines Papers. Für den Fall, dass ein Paper

Datenbank	Für die Suche eingesetzte Queries									
	Q <sub>1</sub>		Q <sub>2</sub>		Q <sub>3</sub>		Q <sub>4</sub>		$\sum$ Datenbank	
	Ges.	Ges.*	Ges.	Ges.*	Ges.	Ges.*	Ges.	Ges.*	Ges.	Ges.**
AISel (KW 9)	26	5	12	4	1	1	6	0	45	5
ACM (KW 7, 8)	415	71	134	39	73	23	64	19	686	75
EBSCO (KW 9)	123	11	27	7	2	0	20	2	172	12
ERIC (KW 9)	9	6	3	1	0	0	0	0	12	6
GS (KW 10)	1030	120	319	70	88	16	145	41	1582	133
IEEE (KW 8, 9)	48	20	18	6	4	3	4	4	74	20
SC (KW 9)	175	129	64	47	8	7	11	7	258	139
SD (KW 9)	154	12	40	5	10	3	13	4	217	12
SL (KW 9)	409	37	109	20	38	7	33	8	589	40
WS (KW 10)	15	12	6	6	0	0	0	0	21	15
$\sum$ (Q <sub>1..4</sub> )	2404	299***	732	156****	224	51****	296	74****	3656	322****

Tabelle 12: Ergebnisübersicht der Literatursuche

\* ohne Duplikate innerhalb einer Datenbank und Query

\*\* ohne Duplikate innerhalb einer Datenbank und aller Queries

\*\*\* ohne Duplikate innerhalb aller Datenbanken und einer Query

\*\*\*\* ohne Duplikate innerhalb aller Datenbanken und aller Queries

zu mehreren Typen zugeordnet werden konnte, entschied der überwiegende Teil des Inhalts über die Einstufung. Tabelle 13 gibt eine Übersicht über die gemäß den Sampling-Richtlinien ausgewählten und der Richtlinien der vorgestellten Dokumentenanalyse (Mayring u. Fenzl, 2014) codierten und anschließend ausgewerteten Arbeiten.

Typ	Anzahl	Arbeiten
Typ 1	11	Anderson et al. (2013), Bahji et al. (2013), Challico et al. (2014), Dicheva et al. (2014), Filippou et al. (2014), Kim u. Lee (2013), Landers (2014), Ng (2013), Rojas et al. (2013), Seaborn et al. (2013), Swacha (2014)
Typ 2	11	Bartel u. Hagel (2014), Berkling u. Thomas (2013), Betts et al. (2013), de Byl (2012), Dominguez et al. (2013), Fabricatore u. López (2014), Fitz-Walter et al. (2011), Jakubowski (2014), Liu et al. (2011), McDaniel et al. (2012), Stöcklin et al. (2014)
Typ 3	11	Ahn et al. (2014), Alves et al. (2014), Ašeriškis u. Damaševičius (2014), de Byl u. Hooper (2013), Dichev et al. (2014), Lewis (2014a), Lewis (2014b), Liu u. Peng (2013), Morford et al. (2014), Nah et al. (2013), Ramirez u. Squire (2014)

Tabelle 13: Übersicht über codierte Arbeiten pro Typ

Die Ergebnisse der Auswertung auf Basis der Treffermenge werden im Folgenden vorgestellt und systematisch aufbereitet. Dieser Abschnitt widmet sich damit den strukturell-inhaltlichen Aspekten von Gamification und bildet so die Wissensbasis für den weiterführenden konzeptuellen Teil dieser Arbeit.

#### 4.2.4 Zwischenergebnis - Dokumentenanalyse

Der Fokus bei der Auswertung liegt vor allem auf den visuellen Elementen, also den Components, da letztlich diese für die Manifestation der darüber liegenden Mechanics und Dynamics gemäß dem GCDP sorgen. Während die Dynamics, wie bereits beschrieben, Ziele einer Lehr-Lerneinheit darstellen, so sind Mechanics als Spielmodi zu verstehen, die sich durch den gezielten Ein-

satz von Components manifestieren.

Bezogen auf die erste Forschungsfrage, konnten auf Basis der ausgewerteten Dokumente die Components *Badge*, *Leaderboard*, *Point*, *Level*, *Quest* und *Unlock* identifiziert werden. Obwohl die Häufigkeit der Nennungen nicht zwingend ein Beleg für deren Wichtigkeit darstellt, kann davon ausgegangen werden, dass die genannten Elemente zu den am häufigsten eingesetzten Game Design-Elementen in Lehr-Lernkontexten zählen. Auch decken sich die gewonnen Erkenntnisse mit Literature Reviews, welche später mit einer vergleichbaren Forschungsfrage durchgeführt wurden, wie beispielsweise die von Schlagenhauser u. Amberg (2015) oder Seaborn u. Fels (2015). Es wurden zudem die Argumentationen von Nah et al. (2013), Ramirez u. Squire (2014) und Werbach u. Hunter (2015) herangezogen, dass es sich bei Achievements um Badges oder andere Rewards handeln kann, wie beispielsweise Trophäen, Ränge, Sterne oder sonstige Awards (Nah et al., 2013, S. 101; Ramirez u. Squire, 2014, S. 640; Werbach u. Hunter, 2015, S. 44). Deshalb wurden diese nicht als separate Components betrachtet, sondern der jeweiligen konkreten Ausprägung zugeordnet. Weitere Components wie z. B. *Gifting* (Swacha, 2014), werden zwar in der einschlägigen Literatur erwähnt, scheinen jedoch aufgrund der Häufigkeit an Erwähnung im Hinblick auf die zu beantwortenden Forschungsfragen nicht im Fokus aktueller Forschung zu sein und sollen deshalb für die Ergebnispräsentation ausgeblendet werden.

Die genannten identifizierten Elemente sollen einzeln bezüglich deren Ausprägungen gemäß der zweiten Forschungsfrage im Folgenden beschrieben und damit die Erkenntnisse aus der Literatur zusammengefasst werden. Die Ergebnisse zu den Subforschungsfragen **F.2.a.**, **F.2.b.** sowie **F.2.c.** werden dabei gemeinsam pro Component aufgeführt. Je Component erfolgt eine kurze Beschreibung. Zudem wurden die identifizierten Attribute, Zusammenhänge und Regeln jeweilig pro Component verdichtet und damit zu allgemeineren Beschreibungen abstrahiert. Diese Verdichtung erfolgte zum einen aufgrund von expliziten Nennungen, beispielsweise eines Attributs und ebenso auf Basis einer Interpretation von codierten Textstellen (Textsegmenten) durch den Forscher. Beispielhaft für Punkte sei dieses Vorgehen in Abbildung 25 illustriert. Die in den Dokumenten gefundenen Textsegmente wurden miteinander verglichen und auf eine erste Abstraktionsstufe abstrahiert. So lässt sich eine Angabe eines Punktes in Form eines Wert-Einheit-Paares abstrakter beschreiben. In einem zweiten Abstraktionsschritt (dieser ist nicht immer möglich oder sinnvoll), können die abstrakten Punkte-Konzepte zu dem Konzept

Punktesystem zusammengefasst werden, da Punkte in einem Systemkontext eine Metrik darstellen, die mehr als einmalige Verwendung findet und deren semantische Bedeutung sich für Nutzer System-weit nicht ändert. Dies führt zu der Beschreibungscharakteristik *System*, was ein Attribut des Konzepts *Point* und damit eine konzeptuelle Anforderung an dieses Konzept darstellt.

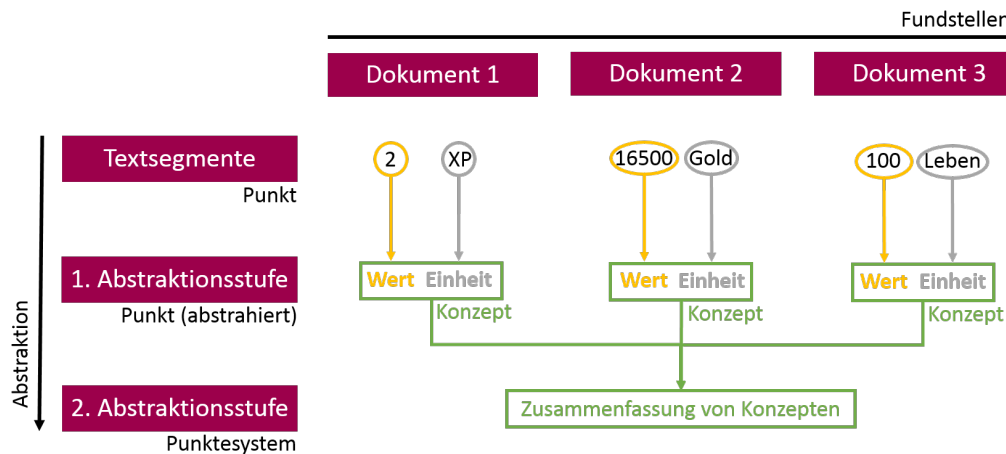


Abbildung 25: Beispielhafte Abstraktion von Textsegmenten zu konzeptuellen Anforderungen von Konzepten

Für die einzelnen Attribute, Zusammenhänge und Regeln werden Referenzen beispielhaft genannt. Dies schließt nicht aus, dass beispielsweise ein Attribut aus mehr als einem Code, stammend aus verschiedenen wissenschaftlichen Materialien verdichtet wurde. Außerdem ist zu beachten, dass die einzelnen Beschreibungen letztlich konzeptuelle Anforderungen im Sinne der später in dieser Arbeit erstellten domänenspezifischen Modellierungssprache darstellen. Deshalb sind die Beschreibungen der Attribute (A), Zusammenhänge (Z) und Regeln (R) mit einer eindeutigen ID versehen.

**BADGES:** Nah et al. beschreibt Badges als „[...] a mark of appreciation or task accomplishment during the process of goal achievement“ (Nah et al., 2014b, S. 405). Badges bedienen das Bedürfnis Anerkennung (Status), können als Lernfortschrittsindikator dienen und spiegeln die Expertise von Lernenden wieder (de Byl, 2012, S. 26). Die Abbildung 26 zeigt ein Beispiel für den Einsatz von Badges in einem e-Learning System.

Badges werden als Motivator für ein bestimmtes Verhalten gesehen. In einem Lehr-Lernkontext können diese intrinsisch motiviertes Verhalten bei Lernenden triggern, wenn Badges von Lernenden verstanden werden als eine Möglichkeit Feedback und Anleitung zu Lernaktivitäten zu bekommen, welche auf ein Ziel hinarbeiten (Ahn et al., 2014, S. 9; Reid et al., 2015, S. 392;





Abbildung 26: Beispiele für Badges in Form von Medaillen in einem LMS (Dominguez et al., 2013, S. 384)

Nah et al., 2014b, S. 405f.). Die extrinsische Motivation kann angesprochen werden, indem sie als Achievement im Sinne eines (Leistungs-) Nachweises in Lernaktivitäten integriert werden (Ahn et al., 2014, S. 9; Reid et al., 2015, S. 392). Welche Motivationsart letztendlich individuell bei Lernenden angesprochen wird, bei der Nutzung eines identischen Badges, kann individuell differieren (Ahn et al., 2014, S. 3, 9). Badges weisen die folgenden Attribute, Zusammenhänge und Regeln basierend auf der vorliegenden Studie auf:

- **B\_A1 Bildliche Repräsentation:** Badges können Stickers oder Icons sein, die selbsterklärend, widerspruchsfrei, nur zu einem Konzept<sup>24</sup> zugeordnet werden und schließlich einen hohen Wiedererkennungswert für den Nutzer besitzen (Alves et al., 2014, S. 560, 566).
- **B\_A2 Titel:** Die textuelle Repräsentation soll kurz, deutlich lesbar und ein Keyword sein, mit dem die bildliche Repräsentation assoziiert ist (Alves et al., 2014, S. 566).
- **B\_A3 Sharing:** Bereits errungene und noch freizuschaltende Badges sollen transparent gemacht werden und die Möglichkeit bestehen, Errungenschaften in sozialen Netzwerken zu verteilen (Alves et al., 2014, S. 562, 566).
- **B\_A4 Feedback:** Lernende sollen über den Erhalt von Badges informiert werden können (Ramirez u. Squire, 2014, S. 643).
- **B\_Z1 Transparenz:** Für den User muss erkennbar sein, mit welchen Aktivitäten der potentielle Erhalt von Badges verknüpft sind (Alves et al., 2014, S. 566). Die Transparenz kann weiter gesteigert werden, indem Nutzern noch nicht freigeschaltete Badges entsprechend mit einer visuellen Markierung angezeigt werden (Dominguez et al., 2013, S. 383).

<sup>24</sup> Als Konzept wird hier eine Kausalität angenommen, welche zur Vergabe eines Badges geführt hat.

Nicht vorhersehbare Badges sind umstritten und es werden positive (Lewis, 2014a, S. 38), wie auch negative (McDaniel et al., 2012, S. 57) Einsätze berichtet und deshalb an dieser Stelle nicht weiter beachtet.

- **B\_Z2 Badges in Leaderboards:** Badges sollen in Leaderboards angeordnet werden können, beginnend mit dem User mit den meisten Badges (Ašeriškis u. Damaševičius, 2014, S. 88).
- **B\_Z3 Badges abhängig von Skill:** Badges können einen Skill, bzw. aus pädagogischer Sicht, ein (Lern-) Ziel oder sogar eine Note repräsentieren oder diesen jeweilig zugeordnet sein (Ašeriškis u. Damaševičius, 2014, S. 89; de Byl, 2012, S. 26; de Byl u. Hooper, 2013, S. 227; Dichev et al., 2014, S. 95).
- **B\_Z4 Badges abhängig von Level:** Badges können ein Level repräsentieren oder diesem zugeordnet sein (Ašeriškis u. Damaševičius, 2014, S. 89).
- **B\_Z5 Badges abhängig von Badges:** Badges können aufsteigend angeordnet werden, so dass ein Badge y nur freigespielt werden kann, wenn der davor liegende Badge x bereits erspielt wurde (Dominguez et al., 2013, S. 383).
- **B\_Z6 Badges abhängig von Points:** Badges können einen Punktestand / Punkteintervall repräsentieren oder diesem zugeordnet sein (Nah et al., 2013, S. 100).
- **B\_R1 Vergabe als Belohnung bei Lernaktivitäten:** Badges können als unmittelbare Folge (z. B. Feedback aufgrund einer erfolgreichen Absolvierung) oder während Lernaktivitäten (z. B. wenn ausreichend Ressourcen gesammelt wurden) vergeben werden (Ahn et al., 2014, S. 4; de Byl, 2012, S. 26; Dominguez et al., 2013, S. 383; Lewis, 2014a, S. 37).

**LEADERBOARDS:** Leaderboards werden dafür verwendet, einen Wettbewerbscharakter bei Lernaktivitäten zu etablieren, indem ein meist kumulierter quantitativer Vergleich von Achievements von einzelnen Lernenden (meist nur ein Ausschnitt der besten Ergebnisse, um Demotivation der niedrigeren Ränge zu vermeiden (Nah et al., 2014b, S. 406)) zum Einsatz kommt (Zichermann u. Linder, 2013, S. 21). Umfrageergebnisse zu den Motivationseffekten durch Leaderboards in Lehr-Lernkontexten lassen annehmen, dass diese zu den motivierendsten Game Design-Elementen zählen (siehe z. B. O'Donovan

et al. (2013)). Die folgenden Attribute, Zusammenhänge und Regeln konnten aus der untersuchten Literatur synthetisiert werden:

- **LB\_A1 Sortierung:** Ein Leaderboard soll anhand von (kumulierten) Achievements (in der Regel Points), absteigend oder aufsteigend sortiert, eine geordnete Liste von Spielern anzeigen (Alves et al., 2014, S. 565).
- **LB\_A2 Repräsentationsformen:** Ein Leaderboard soll den Namen und das Achievement nach dem sortiert wird bei jedem Spieler anzeigen und damit ohne weitere textuelle Information auskommen (Alves et al., 2014, S. 565). Stattdessen sollen Icons verwendet werden. Dabei soll der Name und das Achievement nach dem sortiert wird angezeigt werden (Alves et al., 2014, S. 565).
- **LB\_A3 Reichweite:** Ein Leaderboard kann limitiert oder offen sein (Bartel u. Hagel, 2014, S. 57). Erstere Variante ist vorzufinden, wenn beispielsweise ein Leaderboard im Rahmen einer Lehrveranstaltung verwendet wird, bei der die Teilnehmerzahl begrenzt ist. Ein nicht limitiertes Leaderboard ist gegeben, wenn die Lehrveranstaltung über die Semester hinweg angeboten wird und immer neue Studierende im Leaderboard auftauchen. In einem Einzelspielerkontext sollen die Achievements der vergangenen Spielrunden angezeigt werden (Seaborn et al., 2013, S. 108).
- **LB\_Z1 Leaderboard abhängig von Achievements:** Je mehr Achievements (z.B. Points, Badges, Skills, gelöste Aufgaben etc.) ein Lernender besitzt desto höher ist sein Ranking im Vergleich zu Mitlernenden (Alves et al., 2014, S. 565).

**POINTS:** Ein Punktesystem dient dazu Verhalten zu verfolgen, eine Punktzahl für Spieler aktuell zu halten und dabei Feedback zu geben (Zichermann u. Linder, 2013, S. 19ff.). Dabei können Punkte in einem Punktesystem als Belohnungen vergeben oder als eine Form des Zahlungsmittels<sup>25</sup> bzgl. der Erreichung eines Ziels (z. B. Leistungsnachweise beim Programmieren lernen (Kumar u. Khurana, 2012)) verwendet werden (Nah et al., 2014a). Die Literatur ergab die folgenden strukturellen Beschreibungscharakteristika:

<sup>25</sup> In diesem Fall sprechen beispielsweise Zichermann u. Linder von *Redeemable Points*, also ein Typ von Points, bei dem Points verdient und wieder ausgegeben werden können (Zichermann u. Linder, 2013, S. 19).

- **P\_A1 System:** Points sollen als System von Points auftreten, d.h. nur ein Systemtyp<sup>26</sup> (z. B. Experience Points (Ramirez u. Squire, 2014) oder Karma Points (Seaborn et al., 2013)) soll gleichzeitig eingesetzt werden für eine Anwendung (Alves et al., 2014, S. 560; Chalco et al., 2014, S. 405; de Byl u. Hooper, 2013, S. 222).
- **P\_A2 Feedback:** Lernende sollen über eine Veränderung ihres Punktestandes informiert werden können (Alves et al., 2014, S. 564).
- **P\_A3 Sharing:** Der Punktestand soll durch Lernende in sozialen Netzwerken geteilt werden können (Alves et al., 2014, S. 564).
- **P\_A4 Repräsentation:** Die textuelle Repräsentation soll kurz und deutlich lesbar sein (Alves et al., 2014, S. 564). Wenn stattdessen Icons eingesetzt werden, sollen diese selbsterklärend sein (Alves et al., 2014, S. 564).
- **P\_Z1 Points abhängig von Level:** Mit jedem Levelaufstieg sollen Lernende Points erhalten können (Berkling u. Thomas, 2013, S. 527).
- **P\_Z2 Points durch Hilfe:** Wenn Lernende anderen helfen (z. B. in dem sie Diskussion kommentieren (Betts et al., 2013, S. 217)), soll Helfenden Points vergeben werden können (Berkling u. Thomas, 2013, S. 527).
- **P\_Z3 Points abhängig von Schwierigkeitsgrad:** Je höher der Schwierigkeitsgrad einer Lernaktivität, desto höher soll die Anzahl an Punkten sein, die mit der erfolgreichen Absolvierung der Aktivität vergeben werden (Jakubowski, 2014, S. 252).
- **P\_R1 Belohnung mit Points:** Beim Bearbeiten (z. B. Kommentieren von Diskussionen, schlicht Ansehen, Wiederholen nach Misserfolg (Betts et al., 2013, S. 222) oder Abschließen (de Byl, 2012, S. 27)) von Lernaktivitäten, also durch das Zeigen von gewünschtem Verhalten im System, sollen diese mit Points belohnt werden können (Alves et al., 2014, S. 560, 564).

**LEVELS:** Levels, ebenfalls wie Points angeordnet in einem System, zielen darauf ab, Spielenden ein Gefühl des Fortschritts und der Erfüllung zu vermitteln (Nah et al., 2014a, S. 405). Das Levelsystem teilt dabei den Points-Bereich

<sup>26</sup> Weitere Arten von Points sind natürlich existent. In der ausgewerteten Literatur waren neben Experience und Karma Points noch die folgenden Typen zu finden: Performance Points (Rojas et al., 2013), Viewing Points (Betts et al., 2013), sowie Virtual Currency (Liu et al., 2011).

in kleinere Einheiten, bei dem jede Einheit einem Level zugeordnet ist. Diese strukturierten Fortschrittshierarchien werden in der Regel aufsteigend durchlaufen, an Points bemessen und stellen einen starken intrinsischen Motivator dar, der jedoch auch extrinsisch als Belohnung eingesetzt werden kann (Zichermann u. Linder, 2013, S. 20). Zichermann u. Linder sagen Levels eine große Ähnlichkeit zu Badges nach (Zichermann u. Linder, 2013, S. 20).

- **L\_A1 Progressives Freischalten** Level sollen progressiv freigeschaltet werden können, evtl. fachlich gebunden an eine Taxonomie aus der kognitiven Domäne<sup>27</sup> (Berkling u. Thomas, 2013, S. 527). Die Schwierigkeit der Herausforderungen kann mit dem Levelanstieg steigen (de Byl, 2012, S. 26).
- **L\_A2 Repräsentationsformen:** Ein Level repräsentiert eine diskrete Unterteilung, einen Abschnitt einer Handlung, eine Menge an Herausforderungen oder eine Menge an Ressourcen in einer Spielumgebung (de Byl, 2012, S. 26).
- **L\_A3 Feedback:** Lernende sollen über die Erreichung eines Meilensteins (z. B. ein Levelaufstieg) und ihr aktuelles Level informiert werden (Bartel u. Hagel, 2014, S. 57).
- **L\_Z1 Levelaufstieg abhängig von Points:** Je mehr Points ein Lerner besitzt, desto höher kann sein Level werden (Betts et al., 2013, S. 217; de Byl, 2012, S. 26). Levels hängen demnach kausal von Points ab und besitzen damit einen ähnlichen Belohnungscharakter wie Points.
- **L\_Z2 Unlock abhängig von Level:** Mit Levelaufstiegen soll es möglich sein, neue Unlocks freizuschalten (Betts et al., 2013, S. 217).
- **L\_Z3 Levelaufstieg abhängig von Qualitygate:** Um ein Level aufzusteigen soll es möglich sein, dass ein „end-of-level gate“ eingefügt wird, um das Verständnis des Erlernten zu überprüfen und die notwendigen Punkte für einen Levelaufstieg zu bekommen (Betts et al., 2013, S. 218).

**QUESTS:** Nach Dormans können Quests (auch bekannt als Missionen) drei Zustände haben. Sie können verfügbar (alle Vorbedingungen erfüllt) und damit für den Spieler ausführbar sein. Sie können nicht verfügbar (nicht alle Vorbedingungen erfüllt) und damit nicht ausführbar sein, sowie abgeschlossen,

<sup>27</sup> Berkling u. Thomas (2013) nennen hier die Bloom'sche Taxonomie (Bloom et al., 1956). Denkbar wären auch die überarbeitete Bloom'sche Taxonomie (Anderson u. Krathwohl, 2001) oder die aus dem EVELIN-Projekt stammende Taxonomie von Claren u. Sedelmaier (2012).

was bedeutet, dass ein Spieler einen Quest erfolgreich ausgeführt hat und damit verbundene Folgen zum Tragen kommen (Dormans, 2012, S. 117). Quests sind in der Regel kontextsensitive Bestandteile einer Story (bzw. Narrative oder Storyline) und können je nach pädagogischem Ziel einen Realweltbezug besitzen (Nah et al., 2014a, S. 406).

- **Q\_A1 Kurzüberblick:** Quests sollen einen Kurzüberblick über deren Inhalt und mögliche Optionen bei der Bearbeitung erlauben (Fabricatore u. López, 2014, S. 113).
- **Q\_A2 Instruktion:** Quests beinhalten Instruktionen in Form von Text, Bildern sowie Videos, welches Ziel mit der Erfüllung erreicht werden soll und was dafür getan werden muss (welche Konditionen dafür erfüllt sein müssen) (Alves et al., 2014, S. 565; Fabricatore u. López, 2014, S. 113; Stöcklin et al., 2014, S. 271).
- **Q\_A3 Feedback:** Lernende sollen darüber informiert (textuell oder grafisch) werden, in welchem Zustand sich ein Quest befindet (Alves et al., 2014, S. 566) und wann ein Quest abgeschlossen ist (Fabricatore u. López, 2014, S. 113).
- **Q\_A4 Freiwilligkeit:** Quests können übersprungen werden bzw. müssen nicht zwangsläufig bearbeitet werden (de Byl, 2012, S. 27).
- **Q\_A5 Mehrfache Bearbeitung:** Quests sollen mehrfach bearbeitet werden können, um das Erlernte zu trainieren (Nah et al., 2013, S. 103).
- **Q\_Z1 Quests als Teil von Erzählungen:** Quests, die eingebettet in Erzählungen bzw. Erzählstränge (*“narrative wrappers“* (Ramirez u. Squire, 2014, S. 645)) werden, sind durch die Situiertheit sinnstiftend (*meaningful*) und werden auch als eine Art Meta-Game angesehen, welche bisherige Lernaktivitäten neu für Lernende positionieren (Ramirez u. Squire, 2014, S. 645). Sie können durch alternative Pfade durchlaufen werden (Fabricatore u. López, 2014, S. 113).
- **Q\_Z2 Quests können vorgeschlagen werden:** Ein System soll die Möglichkeit bieten, Quests den Lernenden vorzuschlagen (Alves et al., 2014, S. 566).
- **Q\_Z3 Feedback von Dozierenden:** Zu Questlösungen, die nicht automatisch beantwortet werden können, soll die Möglichkeit bestehen, Feedback durch Dozierende einzuholen, gemäß „[...] scripts for checking“

(Swacha, 2014, S. 198), also vordefinierten Kriterien (Ramirez u. Squire, 2014, S. 645).

- **Q\_R1 Belohnung von Quests:** Nach Abschluss eines Quests sollen Studierende dafür mit Rewards belohnt werden können (Landers, 2014, S. 753; Nah et al., 2013, S. 103).

**CONTENT UNLOCKING:** Im Gegensatz zu dem Freischalten von Achievements zielt das Content Unlocking auf das Freischalten von neuen Lerninhalten bzw. Lernaktivitäten ab (Betts et al., 2013, S. 218). Für das Freischalten von Inhalten gelten vergleichbare Gegebenheiten wie bei dem Freischalten von Achievements: Definierte Vorbedingungen (z. B. Übersteigen einer gewissen Punktzahl oder das Zeigen eines gewünschten Verhaltens im System) müssen erfüllt sein.

- **CU\_A1 Feedback:** Das Freischalten eines neuen Contents soll entsprechend visuell bekannt gemacht werden (Betts et al., 2013, S. 218).
- **CU\_A2 Transparenz:** Unlocks sollen auf einer Profilseite oder anderweitig transparent gemacht werden, sowohl für Spieler als auch für die Community (Ramirez u. Squire, 2014, S. 637).
- **CU\_Z1 Unlock abhängig von Points:** Je mehr Points ein Lernender besitzt, desto mehr Unlocks sollen freigeschaltet werden können (Betts et al., 2013, S. 218).
- **CU\_Z2 Unlock abhängig von Lerneinheiten:** Um spezifische Unlocks freischalten zu können, bedarf es der Fertigstellung von bestimmten „[...] assignments and tasks in order to 'unlock' other challenges“ (Dichev et al., 2014, S. 95).

#### 4.2.5 Maßnahmen zur Qualitätssicherung

Um die Qualität<sup>28</sup> der durchgeführten Dokumentenanalyse zu sichern, wurden, wie in der sozialwissenschaftliche Methodenlehre üblich, diverse Maßnahmen ergriffen. Mit einer sorgfältigen Dokumentation aller Arbeitsschritte und der dadurch geschaffenen Transparenz kann ein grundlegender Anspruch an qualitativ orientierte Forschungsarbeiten erfüllt werden (Flick, 2014,

<sup>28</sup> Klassische Qualitätskriterien, wie Objektivität, Reliabilität und Validität sind in der Literatur vielfach kritisiert (Mayring, 2015, S. 125; Flick, 2014, S. 412ff.), weshalb sich an den Qualitätskriterien von Krippendorff (1980, S. 158) orientiert werden soll.

S. 422) (Mayring, 2015, S. 125) (Kuckartz, 2012). Auf Basis dessen, wurde zudem die Interkoderreliabilität<sup>29</sup> dadurch gesteigert, in dem jeder einzelne Kode durch eine zweite Person gereviewed wurde. Flick bezeichnet dieses Vorgehen als Expertenvalidierung<sup>30</sup> und meint die „[...] Validierung von Ergebnissen durch andere Forschende, denen sie mit Bitte um Kommentierung und Bewertung vorgelegt werden“ (Flick, 2014, S. 415). Bei Abweichungen bezüglich der subjektiven Interpretationsleistung wurde im Gespräch ein Konsens gefunden, welcher schließlich über eine etwaige Rekodierung des jeweiligen Kodes entschied. Zudem wurde ein Beitrag zur semantischen Gültigkeit (Krippendorff, 1980, S. 158ff.) (Validität) geleistet, in dem die zuvor definierten Kategorien und deren Ankerbeispiele sowie Kodierregeln, beim Review mit allen Kodes abgeglichen wurde. Diesem vergleichsweise klassischen Vorgehen zur Sicherstellung der Reproduzierbarkeit (Beitrag zur Steigerung der Reliabilität) und Validität attestieren Flick und Mayring gleichermaßen eine hohe Eignung für das vorliegende Verfahren, da derzeit keine offiziellen Benchmarks existieren, welche den Grad an Zustimmung einer Zweitperson objektiv messbar machen, der benötigt wird, um zu beurteilen, ob eine Aussage valide ist oder nicht (Flick, 2014, S. 416). Bei dem Verfahren zur Beurteilung der Interkoderreliabilität, so Mayring, wird eigentlich das Gütekriterium Objektivität gemessen, „[...] also die Unabhängigkeit der Ergebnisse von der untersuchenden Person“ (Mayring, 2015, S. 124). Im Gegensatz zur Interkoderreliabilität beschreibt die Intracoderreliabilität die Übereinstimmung der Kodes, wenn ein Kodierer nach Abschluss seiner Analyse das Material noch ein Mal kodiert und dabei die Ergebnisse des ersten Durchlaufs nicht kennt (Mayring, 2015, S. 124). Eine vollständige Rekodierung und damit die Erfüllung des Gütekriteriums Stabilität (Krippendorff, 1980) schien aufgrund des damit verbundenen Aufwands nicht zumutbar und ist nach Mayring auch nicht üblich (Mayring, 2015, S. 124). Als ein weiteres Kriterium für Validität wird von Krippendorff die Stichprobengültigkeit genannt (Krippendorff, 1980, S. 158). Wie bereits bei der Beschreibung der Stichprobenauswahl beschrieben, kann die Zufallsauswahl gepaart mit einer Quotenauswahl durchaus kritisch gesehen werden. Dem gegenüber gestellt wird jedoch die Vielfalt an Ergebnissen, welche mit einem erhöhten Maß an Unvoreingenommenheit erzielt werden kann. Auch die korrelative Gültigkeit (Krippendorff, 1980) konnte durch den Vergleich der Ergebnisse mit späteren Untersuchungen, mit ähnlichen Fragestellungen, sichergestellt werden. Vor allem in Bezug auf die erste Forschungsfrage, las-

29 Bei der Interkoderreliabilität wird die Übereinstimmung des Codierungsergebnisses von mindestens zwei verschiedenen Codierern gemessen (Mayring, 2015, S. 124).

30 Mayring spricht in diesem Zusammenhang auch von *Kodierkonferenzen*, bei denen Kodes mit anderen Kodierern diskutiert werden (Mayring, 2015, S. 125).



sen sich vergleichbare Ergebnisse in weiteren Studien ausfindig machen. Die Vorhersagegültigkeit (Krippendorff, 1980) findet in dieser Studie keine Anwendung, da keine Voreingenommenheit der durchführenden Person bestand und demnach keine Ergebnisvorhersagen getätigt wurden. Die Konstruktvalidität (theoretische Gegenstandsorientierung des Forschungsansatzes) (Krippendorff, 1980) konnte sichergestellt werden, indem eine starke Theoriegeleitetheit der Untersuchung vorliegt. Dies ist unter anderem dadurch gegeben, dass ein etabliertes Game Design Framework als Grundlage für die Hauptauswertungskategorien genutzt wurde und darüber hinaus etablierten Prozessempfehlungen (wie z. B. bei der Literatursuche) aus einschlägiger Literatur gefolgt wurde. Das Kriterium der Exaktheit (Krippendorff, 1980) wurde nicht überprüft, da dafür das Qualitätsmerkmal Stabilität erfüllt sein muss, was die Existenz von Intracoderreliabilität voraussetzt.

#### 4.3 LERNPLATTFORMANALYSE

Die Dokumentenanalyse lieferte wertvolle Erkenntnisse für die weitere Forschungsarbeit, jedoch nicht die gewünschte Detaillierung der Game Design-Elemente zur subjektiv-befriedigenden Beantwortung der zweiten Forschungsfrage. Da die einschlägige Fachliteratur vor allem eine sehr wissenschaftliche Sicht repräsentiert und dabei teilweise nur rudimentär bzw. oberflächlich Informationen zu der zweiten Forschungsfragen auftraten, wurde die Analyse auf bestehende spielifizierte Lehr-Lernplattformen<sup>31</sup> ausgeweitet. Damit wurde die wissenschaftliche Sicht durch eine praktische Perspektive ergänzt, was zur Folge hatte, dass die durch das Literature Review bestehende Datenbasis der identifizierten GDE verfestigt werden konnte und neue Perspektiven auf GDEs auftraten. Diese Ausweitung folgt den Argumentationen von iterativ-inkrementeller Artefaktentwicklung nach Hevner et al. (2004), unterstützt durch Patton, welcher sich klar für eine pragmatische und fallgetriebene Anpassung des methodischen Designs im Kontext von strukturierenden Dokumentenanalysen ausspricht, um einer Forschungsfrage in zufriedenstellender Ausführlichkeit zu begegnen:

„[...] the gold standard should be methodological appropriateness rather than methodological orthodoxy [...]. Methodological appropriateness means that designs should be judged on the extent to

31 Unter einer Lernplattform wird in dieser Arbeit ein LMS verstanden und die Definition von Kuhlmann u. Sauter (2008, S. 90) zu Grunde gelegt: „Ein LMS ist eine virtuelle Lern- und Kommunikationsplattform, die den [Lernenden] Zugriff auf verschiedene Lernelemente, z. B. Web-Based Trainings, Dokumente oder Beiträge der [Lernenden], sowie differenzierte Kommunikationsmöglichkeiten bietet“.

which they answer the inquiry question at hand, not whether they adhere to some preordinate standard“ (Patton, 2006, S. i).

Im Folgenden sollen die für die Lernplattformanalyse durchgeführten Schritte beschrieben werden. Dabei ist anzumerken, dass die Lernplattformanalyse nach Abschluss der Dokumentenanalyse begonnen wurde und deshalb die Lernplattformanalyse auf den Ergebnissen der Dokumentenanalyse aufbaut. Deshalb wurde bei der Durchführung darauf geachtet, dass sich die einzelnen Rahmenbedingungen, soweit übertragbar, aus Gründen der Vergleichbarkeit und Homogenität an denen der Dokumentenanalyse orientieren.

#### 4.3.1 *Identifikation von spielifizierten Lernplattformen*

Im Vorfeld der Kriteriendefinition wurde eine systematische Onlinesuche nach spielifizierten Lehr-Lernplattformen durchgeführt, welche insgesamt 38 verschiedene Plattformen aus den unterschiedlichsten Fachdomänen, Sprachen und Bildungsniveaus identifizierte. Die Suche und anschließende Kriterienbasierte Auswahl wurde im Zeitraum April bis Mai 2016 durchgeführt.

#### 4.3.2 *Inklusionskriterien*

Als Datenbasis für die Ausweitung der Untersuchung wurde anhand der folgenden Inklusionskriterien eine Auswahl an Lehr-Lernplattformen getroffen. Die Inklusionskriterien sind an denen des Literature Reviews angelehnt.

- Die Sprache der Lehr-Lernplattform ist Deutsch oder Englisch.
- Die Inhalte der Lehr-Lernplattform besitzen einen erkennbaren Bezug zur Domäne der Informatik. Damit sind jene Plattformen eingeschlossen, die neben Informatikinhalt auch Lerninhalte aus weiteren Themengebieten anbieten wie z.B. der Sprachwissenschaft oder Mathematik.
- Die Lehr-Lernplattform oder zumindest wesentliche Teile daraus sind nicht kommerziell zugangsbeschränkt.
- Die Zielgruppe der Lehr-Lernplattform ist nach eigenen Angaben der Plattform mindestens die tertiäre<sup>32</sup> Bildungsstufe.

---

<sup>32</sup> Im Sinne des Bildungswesens in der Bundesrepublik Deutschland (Kultusministerkonferenz, 2014).

Demnach wurden die in Tabelle 14 gelisteten Lehr-Lernplattformen mit in die Untersuchung einbezogen. Die Angaben in der Tabelle beziehen sich auf das in der Quelle genannte letzte Abrufdatum.

<b>Nr.</b>	<b>Name</b>	<b>Sprache</b>	<b>Anzahl Mit- glieder</b>	<b>Quelle</b>
1	ALISON	Englisch	6 Mio.	Capernaum Ltd. (2016)
2	Classgame	Deutsch	k.A.	Fuchs (2016)
3	Codecademy	Englisch	25 Mio.	Ryzac (2016)
4	Code School	Englisch	1 Mio.	Code School LLC (2016)
5	Coursera	Englisch	18 Mio.	Coursera Inc. (2016)
6	Iversity	Deutsch/ Englisch	k.A.	iversity GmbH (2016)
7	Khan Academy	Deutsch/ Englisch	23 Mio.	Khan Academy Inc. (2016)
8	Open Learning Initiative	Englisch	k.A.	Stanford University (2016a)
9	Lagunita	Englisch	k.A.	Stanford University (2016b)
10	Udacity	Englisch	2 Mio.	Udacity Inc. (2016)
11	Udemy	Deutsch/ Englisch	10 Mio.	Udemy Inc. (2016)

Tabelle 14: Untersuchte spielifizierte Lehr-Lernplattformen

#### 4.3.3 Schema zur Untersuchung von Gamification Design-Elementen

Vor allem hinsichtlich der teilweise sehr kontextspezifischen Ausprägung dieser Elemente, soll zunächst ein Schema<sup>33</sup> entwickelt werden, anhand dessen eine detaillierte Beschreibung der Elemente hinsichtlich ihrer Attribute, Zusammenhänge und Regeln erfolgen kann. Dies trägt dazu bei, eine möglichst hohe Trennschärfe bzgl. der verwendeten Begrifflichkeiten zu etablieren. Es standardisiert zudem die Auswertung indem es sich, wie bei der qualitativen Inhaltsanalyse, an einem Kriterienkatalog orientiert und damit die Vergleichbarkeit der GDE verbessert. Demnach wurden für die Untersuchung die folgenden Kriterien bzw. Aspekte zu Grunde gelegt oder untersucht:

- **Bezeichnung:** Die Bezeichnung orientiert sich an dem bereits vorgestellten Framework von Werbach u. Hunter (2012). Wenn Synonyme auftreten, d.h. GDE die aufgrund ihrer Funktionalität und Eigenschaften einen hohen Deckungsgrad zu einem GDE aus dem Framework aufweisen, so wird es Letzterem zugeordnet und das Synonym kenntlich gemacht.
- **Eigenschaften:** Die Eigenschaften (oder Attribute) eines GDE beschreiben eine statische Sicht auf ein derartiges Element. Beispielsweise können hier typische Bestandteile auftauchen. Vergleichbar ist diese Form der Analyse mit den Ansätzen aus der objektorientierten Analyse, wobei ein GDE einem zu analysierenden Objekt entspricht. Es wird weiterhin unterschieden in Bezeichnung und Ausprägung einer Eigenschaft.
  - Bezeichnung: Name der Eigenschaft.
  - Ausprägungen: Werte oder Werttypen, die eine Eigenschaft annehmen kann.
- **Funktionen:** Mit einer Erfassung von Funktionen wird einer dynamischen Sicht auf ein GDE Rechnung getragen. Hierunter fallen alle vom Benutzer ausführbare Interaktionsmöglichkeiten sowie Funktionalität, die vom GDE selbst ausgeführt wird. Es finden sich hier außerdem die zu erfassenden Regeln wieder.
- **Beziehungen zu anderen GDE:** Die Beziehungen bzw. Zusammenhänge zu anderen GDE werden lediglich dahingehend erfasst, indem benannt wird, mit welchem GDE eine Beziehung existiert und von welcher

---

<sup>33</sup> Das dafür verwendete Schema befindet sich im Anhang A.2 der Arbeit.

Art die Beziehung ist. Hierfür wird eine sprachliche Notation verwendet, die bei der Notation von Multiplizitäten in der Unified Modeling Language (UML) wiederzufinden ist.

- Beziehung zu GDE: Name eines GDE, welches mit dem zu charakterisierenden GDE in Beziehung steht.
- Art der Beziehung: Benennung der Multiplizitäten einer Beziehung.  
Beispiel: *Ein Leaderboard besteht aus 1..\* Punkteständen.*

- **Verwendungskontext:** Der Verwendungskontext beschreibt typische Einsatzfelder des untersuchten GDE. Damit soll der Zusammenhang zu Objekten des Kontextes weiter konkretisiert und darüber hinaus mögliche Regeln identifiziert werden.

- Beschreibung: Erläuterung des Verwendungskontext in konkreten (Lern-) Situationen.

Beispiel: *Nach dem vollständigen und korrekten Bearbeiten einer Erarbeitungsaufgabe, erhält der Benutzer ein Leistungsbadge als eine Form eines Achievements.*

- Taxonomiestufe: Sofern ersichtlich wird hier die Zuordnung eines GDE zu einem Lernziel<sup>34</sup> erfasst, welche in einem Verwendungskontext sichtbar wurde. Zu Grunde gelegt wird die EVELIN-Taxonomie (Claren u. Sedelmaier, 2012), welche in die Taxonomiestufen *Erinnern, Verstehen, Erklären, (angeleitetes) Verwenden, (selbstständiges) Anwenden* und *(Weiter-) Entwickeln* unterteilt (Claren u. Sedelmaier, 2012, S. 648ff.). Die einzelnen Kompetenzniveaus der Taxonomie sind dabei jedoch nicht hierarchisch<sup>35</sup> zu sehen, können jedoch Abhängigkeiten zu anderen Niveaus aufweisen (Claren u. Sedelmaier, 2012, S. 650).

Beispiel<sup>36</sup>: *Nach erfolgreicher Teilnahme an der Lehrveranstaltung, sind die Studierenden dazu in der Lage, selbstständig das Command*

<sup>34</sup> Zu Grunde gelegt wird die Definition von Gonschorek u. Schneider (2010, S. 179), die Lernziele verstehen als „[...] Aussagen, die sich auf ein beobachtbares Verhalten [...] von Lernern beziehen, indem sie dieses Verhalten vorwegnehmend (antizipierend) beschreiben“.

<sup>35</sup> Ähnlich wie bei der allgemeinen Kompetenztaxonomie von Anderson u. Krathwohl (2001), sind die Niveaus oder auch Stufen genannt nicht hierarchisch zueinander und geben eine Orientierung sowohl für Lehrende als auch Lernenden, welche Schwierigkeitsgrade operationalisiert in Form von Lernzielen bestehen (Anderson u. Krathwohl, 2001).

<sup>36</sup> Lernziele ergeben sich dabei aus einer sprachlichen Kombination von Adressatenkreis (in der Hochschulbildung Studierende), dem Objekt, was dem fachlichen Inhalt entspricht und der Taxonomiestufe (ausgedrückt durch ein Tätigkeitsverb (Figas et al., 2014, S. 135)). Diese Auffassung basiert auf der behaviouristischen *lernzielorientierten Didaktik* nach Möller (2006, S. 75f.), welche aktiven Tätigkeitsverben, jeweilig zu den sechs Kompetenzniveaus der EVELIN Taxonomie zuzuordnen sind (Figas et al., 2014, S. 135).

*Design Pattern mit undo in für sie neuen Situationen anzuwenden* (Bartel u. Hagel, 2016b, S. 76, eigene Übersetzung).

- Aufgabentypen: Die Beschreibung des Verwendungskontextes komplettiert die Erfassung des Aufgabentyps, sofern dieser ersichtlich ist und eine direkte Verbindung zum GDE erkennen lässt. Dafür soll die in beispielsweise Figas et al. (2015b) und Abraham u. Müller (2009) verwendete und bewährte Unterscheidung zwischen *Lernaufgaben* und *Leistungsaufgaben* zum Einsatz kommen (Figas et al., 2015b).

Lernaufgaben dienen dem Zweck den Kompetenzerwerb zu fördern und dabei Fehler als Chance zu sehen, vorhandene Wissenslücken (je nach Aufgabenstellung kooperativ und kommunikativ mit anderen Lernenden) zu schließen (Figas et al., 2015b; Keller u. Bender, 2012; Seel, 1981). Damit steht das Lernen als solches im Vordergrund, welches durch eine Aufgabenbearbeitung durch Lernende, diese dazu anregt, ihre Kompetenzen zu entwickeln (Figas et al., 2015b, S. 21). Demnach bedarf es bei dieser Art von Aufgabentyp nach Bloemen u. Schlömer (2012) einer Bereitstellung von „[...] gestaltungs- und kompetenzorientierte[n] Lernsituationen“ (Bloemen u. Schlömer, 2012, S. 128). Lernaufgaben können nach Figas et al. weiterhin anhand ihrer didaktischen Funktion<sup>37</sup> unterteilt werden in *Erarbeitungsaufgaben*, *Wiederholungs- und Übungsaufgaben*, *Anwendungsaufgaben* und *freie Gestaltungsaufgaben* (Figas et al., 2015b, vgl. S. 25). Evaluiert werden können Lernaufgaben entweder vom Lernenden selbst oder durch externe Beurteiler.

Leistungsaufgaben hingegen, zielen darauf ab, Wissen und Kompetenzen zu diagnostizieren bzw. das Erreichen von Lernzielen zu überprüfen (ggf. mit Benotung) (Abraham u. Müller, 2009; Maier et al., 2010). Folglich verfolgen Lernende ein Leistungsmotiv, die gestellten Anforderungen zu erfüllen (Abraham u. Müller, 2009, vgl. S. 5). Fehler in diesem Kontext sind als Wissens- oder Kompetenzdefizite anzusehen (Keller u. Bender, 2012). Über Erfolg bzw. Misserfolg entscheidet in der Regel nicht der Lernende selbst.

Die Erfassung dieses Aspekts ist insbesondere dafür wichtig, einen Erkenntnisgewinn darüber zu erlangen, wie Aufgabentypen mit

<sup>37</sup> Für eine ausführliche Diskussion dieser Unterteilung, sei auf Figas et al. (2015b) verwiesen. Außerdem existieren weitere Klassifikationsschemata zur Unterteilung von Aufgaben anhand definierter Kriterien, wie beispielsweise das von Maier et al. (2010).

GDE zusammen agieren können. Darüber hinaus können typische Aufgaben und deren Struktur so erfasst werden. Über eine Kombination von Lern- und Leistungsaufgaben sind in der Literatur sehr heterogene Aussagen und Betrachtungsweisen zu finden. Beispielsweise geben Abraham u. Müller an, dass Leistungsaufgaben nicht wie häufig üblich am Ende eines Lernprozesses verortet sein müssen, sondern auch in Mitten eines Lernprozesses als „Bestandsaufnahmen“ sowohl für Lernende als auch Lehrende fungieren können (Abraham u. Müller, 2009, vgl. S. 7). Dagegen argumentieren beispielsweise Keller u. Bender, welche Lernaufgaben sowohl einen lernförderlichen als auch einen lerndiagnostischen Charakter, wie bei Leistungsaufgaben, zusprechen (Keller u. Bender, 2012, vgl. S. 8)

Weiterhin gilt für die Analyse, dass diese keine Redundanzen enthält. Dies bedeutet, dass beispielsweise Eigenschaften von einem GDE nur einmal erfasst werden, sofern sie im Untersuchungsmaterial mehr als ein Mal in gleicher Gestalt auftauchten.

#### 4.3.4 Durchführung

Der Zugang zu den Plattformen erfolgte über eine kostenlose Registrierung, welche – wie bereits in den Inklusionskriterien definiert – wesentliche Lerninhalte verfügbar machte. Die Ergebnisse dieser Analyse beziehen sich also auf den Korpus an Elementen, der bei dieser Art der Registrierung einsehbar war. Es ist demnach nicht auszuschließen, dass ein Upgrade auf einen zahlungspflichtigen Account, wie es beispielsweise die Plattform *Codeacademy* anbietet, zur Folge hat, dass weitere Elemente zum Vorschein treten können, die in der vorliegenden Analyse keine Berücksichtigung finden konnten.

Die Plattformen wurden in sequentieller Reihenfolge abgearbeitet und pro Plattform alle GDE erfasst. Nach der vollständigen Erfassung wurde aus den daraus entstandenen ausgefüllten Schemata die Relevanten den GDE aus der Dokumentenanalyse zugeordnet. Zudem wurde nach dem Erfassen der ersten GDE mit Hilfe des vorgestellten Schemas bemerkt, dass Konstrukte existieren, welche sich zwar mit vorhandener Schemastruktur beschreiben lassen, jedoch lediglich eine (teilweise hierarchische) Zusammensetzung von anderen GDEs oder Kontextobjekten darstellen und diese unter neuem Namen repräsentieren. Deshalb wurden die Bezeichnungen Component Class (CC) und Lernele-

ment eingeführt. Beispielhaft kann hier eine *CC Profil* angeführt werden, welche Kategorien und/oder Kurse einer Lerneinheit zusammenfasst und Profildaten, wie erspielte Achievements, gesammelt darstellt. Beispiele für Lernelemente sind Single Choice-Aufgaben<sup>38</sup>, Diskussionsforen oder Lernmaterialien, wie beispielsweise Videos oder Quellcode. Diese Elementtypen werden hauptsächlich in dem letzten Abschnitt des Schemas beschrieben, dem Verwendungskontext. Wurden bei einer Lernplattform mehrere Lerneinheiten oder Kurse angeboten, die sich in der Anzahl der eingesetzten GDE, CC oder Lernelemente unterschieden, wurde diejenige bzw. derjenige ausgewählt, welche zahlenmäßig am Meisten Elemente verwendete.

Darüber hinaus ist zu erwähnen, dass für vergleichbare GDE, CC bzw. Lernelemente von den einzelnen Plattformen unterschiedliche Begriffe<sup>39</sup> oder Ausprägungen verwendet werden, alleinig schon aus Gründen der durch die Plattformen angebotenen Sprachen. Diese teils synonymischen Verwendungen von GDE wurden im Rahmen der Analyse den bereits erfassten GDE zugeordnet, sofern eine Zuordnung valide erschien. Für CC und Lernelemente existierte seitens der Dokumentenanalyse kein Input, weshalb bei beiden Konzepten jeweils alle Varianten erfasst wurden, um sie in einem späteren Schritt auf Basis der Schemata vergleichen zu können.

Demnach sollen im Folgenden auch Lernelemente beschrieben werden, um genauere Erkenntnis darüber zu erlangen, wie GDE oder CC mit diesen Elementen zusammenhängen und wie sich diese Elemente in einer späteren Umsetzung in einer Plattformlösung ausgestalten können. Für ein besseres Verständnis wurden zudem die Zusammenhänge von GDE, CC und Lernelemente für jede Plattform visualisiert. Die Abbildung 27 zeigt ein Beispiel für diese strukturelle Übersicht, welche GDE (grün), CC (blau) und Lernelemente (gelb) in UML-ähnlicher Notation anordnet.

Darin ist beispielsweise zu erkennen, dass eine Gesamtpunktzahl in einem Profil angezeigt wird, während die Gesamtpunktzahl für jedes Profil individuell ist. Für die beispielhaft dargestellte Plattform *Classgame* wurde der Übungskurs gewählt, da dieser mehr zu analysierende Elemente enthielt, als der Pflichtkurs.

38 Auch weitere Typen von Aufgaben, unabhängig davon, ob sie als Lern- oder Leistungsaufgabe auftreten.

39 Für alle Plattformen wurde bei der Analyse versucht Deutsch zu verwenden, aus Gründen der Vergleichbarkeit. Wenn eine Plattform ein deutsches Sprachpaket nicht unterstützte, wurde Englisch verwendet.



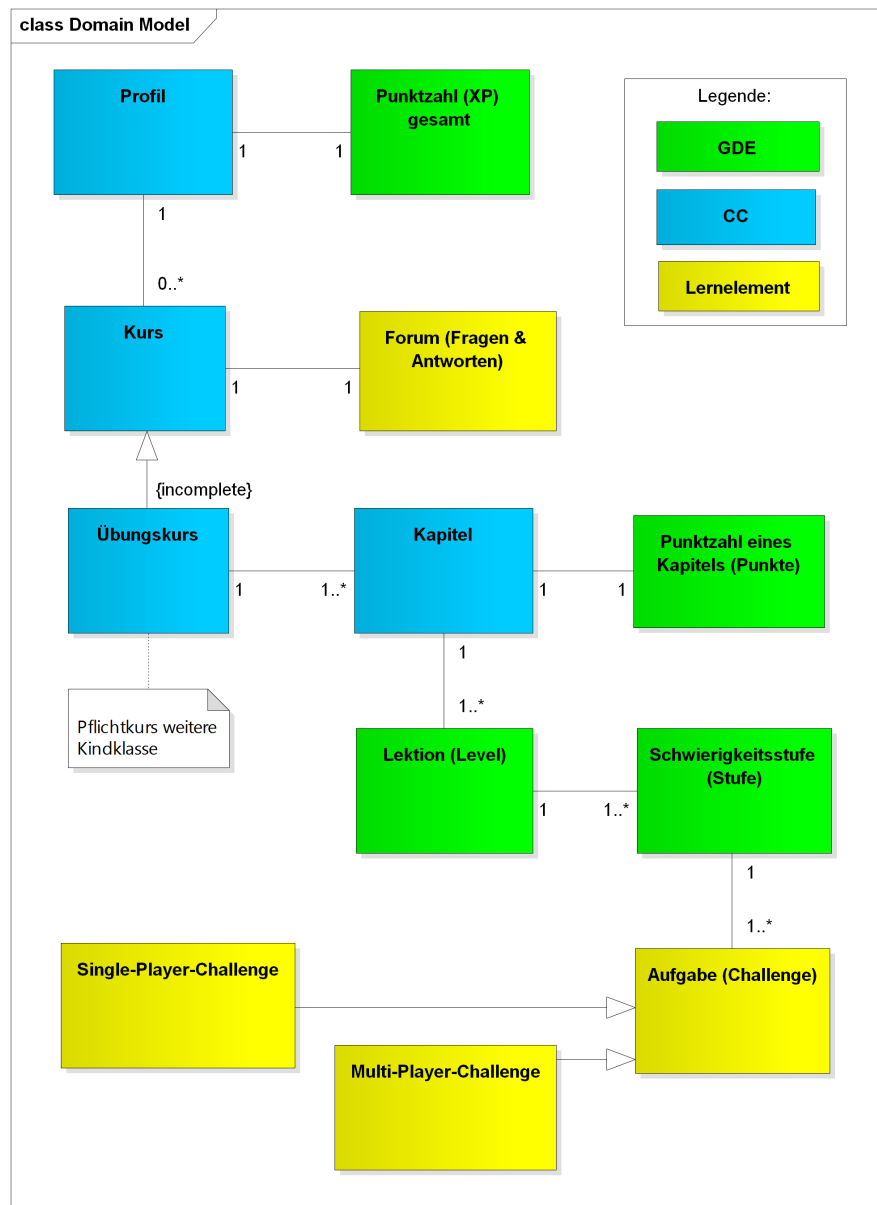


Abbildung 27: Beispielhafte Visualisierung der GDE, CC und Lernelemente in einem Übungskurs der Lernplattform *Classgame* (Fuchs, 2016)

#### 4.3.5 Zwischenergebnis - Lernplattformanalyse

Im Folgenden sollen die Ergebnisse der Lernplattformanalyse aufbauend auf der Dokumentenanalyse vorgestellt werden. Dafür werden die laufenden Nummern aus den Beschreibungskonzepten der Dokumentenanalyse fortgesetzt, sofern ein neues Attribut, ein weiterer Zusammenhang oder eine Ableitung einer Regel respektive eines GDE möglich war. Darüber hinaus werden Elemente der durch die Lernplattformanalyse durchgeführten Konzepte CC und Lernelement aufgeführt und gleichermaßen wie GDE beschrieben. Die Basis

für diese Beschreibungen stellen die Schemata dar, welche miteinander abgeglichen wurden und schließlich das Delta zu den Daten aus der Dokumentenanalyse bestimmt wurde. Wie bereits in der Dokumentenanalyse, wurden mit einer inhaltlichen Verdichtung gearbeitet, um mehrfache Nennungen, teilweise interpretiert durch den Forscher, zu übergeordneten Konzepten zu abstrahieren.

#### 4.3.5.1 *Game Design-Elemente*

##### BADGE

- **B\_A5 Freigeschaltet:** Badges können auf eine Aktion und zu einem bestimmten Zeitpunkt hin freigeschaltet werden oder sind verfügbar, um noch freigeschaltet zu werden (Khan Academy Inc., 2016).

##### POINTS

- **P\_A5 Veränderung:** Die Gesamtpunktzahl kann sich erhöhen, aber beispielsweise bei Falschantworten auch fallen (Capernaum Ltd., 2016). Bei Wiederholung von Aufgaben, welche mit Points belohnt werden, werden die zuvor erspielten Points zurückgesetzt (Capernaum Ltd., 2016).
- **P\_Z4 Points abhängig zu Profil:** Die Gesamtpunktzahl soll in einem Profil für Lernende sichtbar sein (Khan Academy Inc., 2016).
- **P\_Z5 Points abhängig von Abschlussassessment:** Auch wenn nicht jede Lernaktivität mit der Vergabe von Points belohnt wird, so kann eine Reihe von Aufgaben existieren, welche den Abschluss einer Lerneinheit (oder einem Teil davon) bilden und bei denen jeweils bei korrekter Beantwortung Points vergeben werden (Stanford University, 2016a; Stanford University, 2016b).

##### LEVEL:

- **L\_A4 System:** Ähnlich wie Points, sollen Level in einem System auftreten, welches sich durch eine Anwendung durchzieht und einzelne Level mit anderen GDE oder Lernaktivitäten verbunden sind (iversity GmbH, 2016).

**SKILL:** Ein Skill kann mit einem Lernziel gleichgesetzt werden und soll als weiteres GDE, basierend auf der Analyse der Plattform *Codecademy* (Ryzac, 2016) eingeführt werden. Ein Skill ist strukturell ähnlich wie ein Badge.

- **S\_A1 Freischaltet:** Skills sollen bedingt durch eine oder mehrere Aktionen freigeschaltet werden können, beispielsweise nach dem erfolgreichen Abschluss von Teilen einer Lerneinheit (Ryzac, 2016).
- **S\_A2 Feedback:** Lernende sollen darüber informiert werden, wenn sie einen neuen Skill erreichen (Ryzac, 2016).
- **S\_A3 Bezeichnung:** Jeder Skill soll, wie auch Badges, eine eindeutige und sprechende Bezeichnung besitzen (Ryzac, 2016).
- **S\_Z1 Skill abhängig von Lerneinheiten:** Ein Skill repräsentiert eine Zusammenfassung von einer oder mehreren Lerneinheiten (z. B. eines Kurses bestehend aus Lerneinheiten, wobei nach dem Abschluss einer Menge von Einheiten ein oder mehrere Skills damit erworben werden (Ryzac, 2016), siehe auch z. B. Landers et al. (2017, S. 474) oder Kapp (2012b, S. 38)).

#### 4.3.5.2 *Container Classes*

**PROFIL:** Ein Profil (auch Vitrine (Khan Academy Inc., 2016) oder Dashboard (iversity GmbH, 2016) genannt) ist eine CC, welche die unterschiedlichsten, für den Lernenden relevanten Informationen aufbereitet darstellt. Darunter fallen beispielsweise neben einer Übersicht über die persönlichen Daten auch Lernfortschrittsstände oder erspielte GDE.

- **PR\_A1 Persönliche Daten:** Das Profil soll persönliche Daten, wie Benutzerbild, Geschlecht und Name (ggf. Nickname) enthalten (Coursera Inc., 2016).
- **PR\_A2 Zustand Lerneinheiten:** Das Profil soll den aktuellen Lernfortschritt (z. B. Stand von Lerneinheiten, wie abgeschlossene Aufgaben, Gesamtspielzeit oder abgegebene Kommentare zu Diskussionen etc. (Khan Academy Inc., 2016)) wiedergeben (Coursera Inc., 2016; Ude-my Inc., 2016).
- **PR\_Z1 Profil abhängig von eingesetzten GDE:** Bisherige Achievements (z. B. Points, Badges oder Skills) sollen angezeigt werden können (Khan Academy Inc., 2016). Beispielsweise kann der Nutzer bei Khan Academy Inc. auswählen, welche fünf Achievements für ihn am Wichtigsten sind (Khan Academy Inc., 2016).

- **PR\_R1 Profilaktualisierung:** Das Profil eines Lernenden soll immer den aktuellsten Zustand bezogen auf die dort aufbereiteten Informationen enthalten und wird deshalb bei jedem Aufruf aktualisiert (siehe z. B. (Ryzac, 2016)).

**DISKUSSION:** In einem Diskussionsbereich (auch „Fragen und Antworten“ (Udemy Inc., 2016) oder „Forum“ (Udacity Inc., 2016) genannt) können Lernende Diskussionen starten, welche mit Kommentaren beantwortet werden und so eine Form der asynchronen Kommunikation ermöglichen (Kuhlmann u. Sauter, 2008, S. 110).

- **D\_A1 Anzahl Diskussionen:** Der Diskussionsbereich soll die Anzahl der momentanen Diskussionen beinhalten (Udacity Inc., 2016; Coursera Inc., 2016).
- **D\_A2 Anzahl Kommentare:** Der Diskussionsbereich soll die Anzahl der momentanen Kommentare beinhalten (Udacity Inc., 2016; Coursera Inc., 2016).
- **D\_A3 Metadaten:** Jede Diskussion erhält einen Titel, einen Text, einen Erstzeitpunkt sowie eine erstellende Person (Udacity Inc., 2016; Coursera Inc., 2016).
- **D\_A4 Kommentare:** Eine Diskussion besteht aus Kommentaren, die von Lernenden verfasst werden und welche dieselben Metadaten (bis auf den Titel) besitzen wie Diskussionen (Coursera Inc., 2016).
- **D\_A5 Kommentare bewerten:** Kommentare zu Diskussionen können von Lernenden als hilfreich für eine Problemlösung markiert (Khan Academy Inc., 2016) werden und beispielsweise der hilfreichste Kommentar mit Points bewertet werden (Fuchs, 2016).
- **D\_A6 Suche:** Kommentare zu Diskussionen können von Lernenden anhand von Suchbegriffen durchsucht werden (iversity GmbH, 2016).
- **D\_Z1 Diskussion zu Lernelemente:** Bei der Bearbeitung von Lernelementen soll es möglich sein, Fragen zu einem spezifischen Lernelement zu stellen und dadurch Diskussionen zu erstellen (Fuchs, 2016).

**LERNEINHEIT:** Eine Lerneinheit (auch als Kapitel (Fuchs, 2016), Lektion (Coursera Inc., 2016) oder Lesson (Ryzac, 2016) bezeichnet) kapselt Lernbausteine, welche von Lernenden in einer bestimmten Reihenfolge bearbeitet

werden können. Synonyme sind Kapitel (iversity GmbH, 2016) oder Modul (Stanford University, 2016a; Stanford University, 2016b).

- **LE\_A1 Lernziele:** Eine Lerneinheit besitzt für Lernende transparente Lernziele, die mit der Bearbeitung dieser erreicht werden können (Fuchs, 2016).
- **LE\_A2 Bezeichnung:** Eine Lerneinheit besitzt eine Bezeichnung (textuell oder grafisch) (Fuchs, 2016) und ggf. eine Nummerierung (Ryzac, 2016).
- **LE\_A3 Zeitaufwand:** Eine Lerneinheit besitzt eine Schätzung, wie viel Zeit eine vollständige Bearbeitung unter Berücksichtigung der Vorkenntnisse von Lernenden in Anspruch nimmt (iversity GmbH, 2016). Optional kann ein Start- und ein Endtermin für die Bearbeitung angegeben werden (Ryzac, 2016).

#### 4.3.5.3 Lernelemente

Lernelemente sind, wie der Name bereits intendiert, Elemente, die zum Lernen genutzt werden. Darunter fallen neben unterschiedlichen Typen von Aufgaben auch Materialien oder Möglichkeiten der Kommunikation, wie beispielsweise Diskussionsforen. Wichtig ist zu erwähnen, dass die bereits identifizierten Quests inhaltlich spielifizierte Lernelemente darstellen. Es handelt sich also um nicht nur eine rein strukturelle Spielifizierung (Kapp et al., 2014, S. 55f.).

Bei der Lernplattformanalyse konnten die folgenden Lernelemente identifiziert und analysiert werden.

**AUFGABE:** Bei der Analyse kristallisierten sich insgesamt vier unterschiedliche Aufgabentypen heraus (siehe Abbildung 28): Single Choice-Aufgaben (z. B. iversity GmbH (2016)), Multiple Choice Aufgaben (z. B. Coursera Inc. (2016)), Eingabe Aufgaben (z. B. Khan Academy Inc. (2016)) und Freitext Aufgaben (z. B. Fuchs (2016)). Single Choice-Aufgaben (Abbildung 28a) sind Aufgabentypen, bei denen nur eine Antwort aus einer Reihe an Antworten die Fragestellung der Aufgabe korrekt beantwortet. Multiple Choice-Aufgaben (Abbildung 28b) sind vergleichbar mit Single Choice-Aufgaben, nur dass bei diesem Aufgabentypus mindestens zwei Antwortmöglichkeiten eine Aufgabe korrekt lösen. Bei Eingabe-Aufgaben (Abbildung 28c) wird ein Wert oder eine kurze Zeichenkette abgefragt und bei Freitext-Aufgaben (Abbildung 28d) wird ein längerer Text oder Quellcode als Antwort auf eine Frage erwartet.

Da die identifizierten Aufgabentypen sich strukturell hauptsächlich in ihrem Antwortformat unterscheiden, sollen im Folgenden die Attribute, Zusammenhänge und Regeln allgemeiner auf der Ebene von Aufgaben formuliert werden.

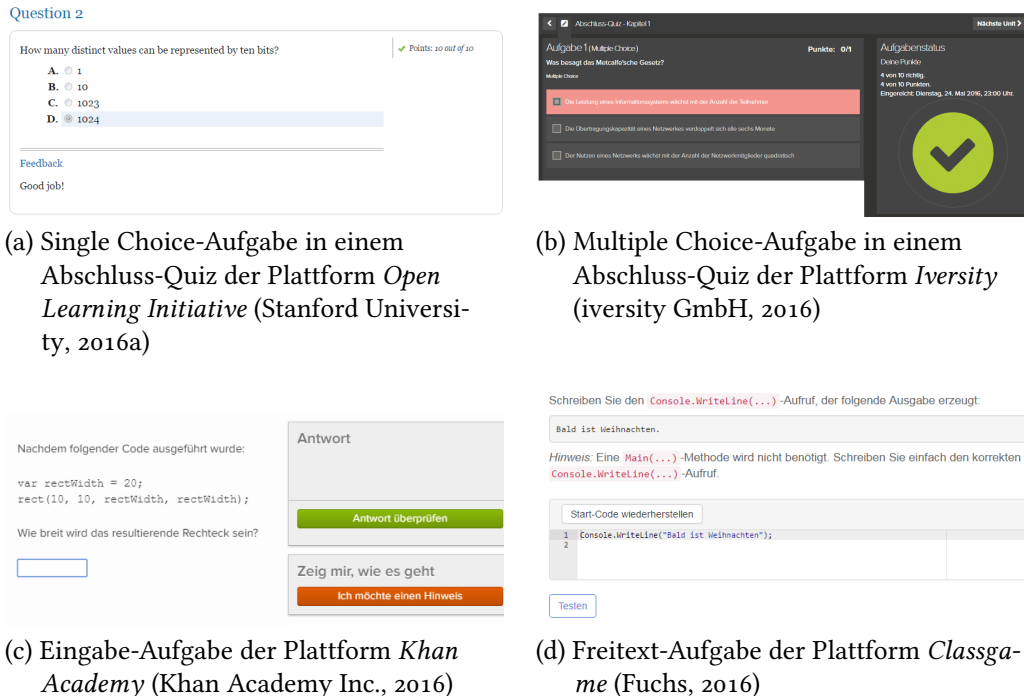


Abbildung 28: Beispielhafte Darstellung der vier identifizierten Aufgabentypen

Die folgenden Beschreibungscharakteristika konnten bei der Analyse identifiziert werden:

- **A\_A1 Titel:** Eine Aufgabe besitzt eine Bezeichnung bzw. einen Titel (Stanford University, 2016a).
- **A\_A2 Fragentext:** Eine Aufgabe besitzt einen Fragentext bzw. eine Aufgabenstellung (Stanford University, 2016a). Ein Fragentext kann mit Hilfe eines Videos oder eine andere mediale Aufbereitung von Lerninhalten unterstützt sein, wie beispielsweise einem Bild oder einem Podcast (Udacity Inc., 2016).
- **A\_A3 Antworten:** Eine Aufgabe besitzt unterschiedliche Antwortformate. Eine Aufgabenlösung kann durch Lernende ausgewählt bzw. abgegeben und ggf. durch andere Lernende oder von Lehrenden manuell sowie automatisch bewertet werden (Fuchs, 2016).
- **A\_A4 Feedback:** Lernende sollen über die Richtigkeit ihrer Antwort informiert werden können (Khan Academy Inc., 2016).

- **A\_A5 Freigeschaltet:** Aufgaben können durch Lernende freigeschaltet werden, indem beispielsweise für die Freischaltung notwendige Aufgaben korrekt gelöst wurden (Fuchs, 2016).
- **A\_A6 Lesezeichen:** Aufgaben können durch Lernende als Lesezeichen markiert und zusätzlich verschlagwortet werden (Fuchs, 2016).
- **A\_A7 Anzahl Versuche:** Sofern eine Mehrfachbeantwortung von einer Aufgabe zulässig ist, soll die Anzahl der Versuche erfasst und transparent dargestellt werden (iversity GmbH, 2016). Evtl. kann eine zeitliche Verzögerung (Cooldown-Phase (Fuchs, 2016)) gefordert werden, bis die Aufgabe erneut bearbeitet werden kann.
- **A\_A8 Hilfestellungen:** Während der Bearbeitung einer Aufgabe sollen Hilfestellungen angezeigt werden können, die einen Beitrag zum Lernerfolg leisten (Ryzac, 2016; Khan Academy Inc., 2016).
- **A\_A9 Frage stellen:** Während der Bearbeitung einer Aufgabe soll die Möglichkeit bestehen, eine Frage zu einer Aufgabe zu stellen und damit eine Diskussion zu starten (Fuchs, 2016).
- **A\_Z1 Aufgabe zu GDE:** Für die korrekte Beantwortung einer Aufgabe können Belohnungen in Form von Achievements erspielt werden (z. B. Khan Academy Inc. (2016)).

#### *Verwendungskontext von Aufgaben*

Interessant ist, dass Anwendungsaufgaben überwiegend im Kontext von Leistungsaufgaben verwendet werden. Derartige Aufgaben, bei denen Lernende fachliche Inhalte angeleitet verwenden oder selbstständig anwenden können, waren vergleichsweise deutlich seltener zu finden. In *Khan Academy* wurde beispielsweise solch eine Form der Aufgabeneinbettung vorgenommen. Die in einem Javascript Kurs angebotene anwendungsorientierte Lernaufgabe stellte eine solche Ausnahme dar, bei der Lernende unter Anleitung versuchen sollen, einen Schneemann, bestehend aus drei Kreisen mit Programmcode zu konstruieren. Auch zeichnete sich ein deutliches Bild bezogen auf die unterschiedlichen Aufgabentypen ab. Die weitverbreitete Verwendung von Single Choice- und Multiple Choice-Aufgaben spricht dafür, dass in erster Linie diese den Zweck von Wiederholung bzw. Übung oder Anwendungsaufgaben erfüllen. Erarbeitungsaufgaben waren deutlich seltener zu finden, wenn dann nur in Verbindung mit Lernmaterialien (oder nur repräsentiert durch Lernmaterialien, bei denen keine Aufgabenstellung vorhanden war), bei denen Lernende

beispielsweise für die Bearbeitung einer Aufgabe ein Video anzusehen hatten. Freie Gestaltungsaufgaben waren hingegen nicht ausfindig zu machen. Diese Feststellungen decken sich mit einer im Vorfeld durchgeführten Analyse, welche untersuchte, welche Aufgabentypen in Übungsblättern des Fachs Software Engineering an deutschsprachigen Hochschulen zu finden waren (siehe hierzu Figas et al. (2015b)). Dabei wurde, ebenso wie bei dieser Analyse, festgestellt, dass ein mengenmäßiger deutlicher Fokus auf dem Einsatz von Anwendungs- sowie Wiederholungs-/Übungsaufgaben liegt und Erarbeitungsaufgaben und freie Gestaltungsaufgaben quasi keine Beachtung in der Lehre finden (Figas et al., 2015b, S. 25f.).

Zudem erscheint es interessant zu erwähnen, dass vornehmlich die unteren Taxonomiestufen, also *Erinnern* und *Verstehen* durch die in den Plattformen identifizierten Aufgaben bei Lernenden angesprochen wurden. Aufgaben, bei denen Lernende Sachverhalte erklären, selbstständig anwenden oder gar weiterentwickeln sollten, waren hingegen nicht zu finden. An dieser Stelle sei nochmals angemerkt, dass nicht ausgeschlossen werden kann, dass dies bei Bezahlvarianten von den angebotenen Kursen sich anders verhalten kann.

#### *Weitere Lernelemente*

Neben Aufgaben konnten noch weitere Lernelemente identifiziert werden, welche im Folgenden vorgestellt werden. Die folgenden Lernelemente dienten vor allem der Wissensaneignung und waren sowohl zu Beginn (z. B. Coursera Inc. (2016)) einer Lerneinheit als auch in der Mitte (z. B. Code School LLC (2016)) dieser zu finden.

**VIDEO:** Lerninhalte werden in Videos grafisch aufbereitet und durch Ton unterstützt.

- **V\_A1 Bezeichnung:** Ein Video besitzt eine Bezeichnung bzw. ein Titel (Khan Academy Inc., 2016).
- **V\_A2 Videoplayer:** Ein Videoplayer mit entsprechenden Steuerungsmöglichkeiten (z. B. abspielen, pausieren etc.) spielt eine Mediendatei ab (Code School LLC, 2016), welche beispielsweise auf einer Plattform wie *Youtube* gehostet ist.
- **V\_A3 Frage stellen:** Während dem Ansehen eines Videos soll die Möglichkeit bestehen, eine Frage zu einem Video zu stellen und damit eine Diskussion zu starten (Udacity Inc., 2016).



- **V\_A4 Freischaltet:** Da Aufgaben durch Lernende freigeschaltet werden können und Videos Teile von Aufgaben sein können (siehe V\_Z2), sollen auch Videos freigeschaltet werden können (Khan Academy Inc., 2016).
- **V\_Z1 Einbettung in Aufgabe:** Videos können in Aufgaben eingebettet sein, beispielsweise als Teil einer Fragestellung (Ryzac, 2016).
- **V\_Z2 Video zu GDE:** Für das Ansehen von Videos können Belohnungen in Form von Achievements erspielt werden (Khan Academy Inc., 2016).

**PODCAST:** Ein Podcast ähnelt einem Video, verzichtet jedoch auf den Einsatz von Bildern und stützt sich stattdessen nur auf Ton (z. B. Radio- oder Fernsehsendungen, Lerntagebücher oder Diskussionswiedergaben (Kuhlmann u. Sauter, 2008, S. 158)). Dieser Medientyp wurde zwar nicht in den untersuchten Lernplattformen gefunden, soll jedoch aufgrund der strukturellen Ähnlichkeit zum Video ebenfalls in den Katalog der konzeptuellen Anforderungen einfließen. Dies begründet sich dadurch, dass beispielsweise Rey (2009, S. 17) dem Podcast beim Lernen eine gewisse Bedeutung zumisst, die in der Emendo DSML berücksichtigt werden soll. Kuhlmann u. Sauter attestieren Podcasts auf Basis der Aussagen von Erpenbeck u. Sauter ein starkes Potential für die Entwicklung von personalen Kompetenzen sowie mittlere Potentiale für die Entwicklung von aktivitätsbezogenen und sozialen Kompetenzen (Kuhlmann u. Sauter, 2008, S. 59, 158).

- **PC\_A1 Bezeichnung:** Ein Podcast besitzt eine Bezeichnung bzw. ein Titel.
- **PC\_A2 Podcastplayer:** Ein Podcastplayer mit entsprechenden Steuerungsmöglichkeiten (z. B. abspielen, pausieren etc.) spielt eine Audiodatei ab.
- **PC\_A3 Frage stellen:** Während der Bearbeitung eines Podcasts soll die Möglichkeit bestehen, eine Frage zu einem Podcast zu stellen und damit eine Diskussion zu starten.
- **PC\_A4 Freigeschaltet:** Podcasts sollen ebenso wie Videos freigeschaltet werden können.
- **PC\_Z1 Einbettung in Aufgabe:** Podcasts können in Aufgaben eingebettet sein, beispielsweise als Teil einer Fragestellung.

- **PC\_Z2 Podcast zu GDE:** Für das Anhören von Podcasts können Belohnungen in Form von Achievements erspielt werden.

**TEXTBLOCK:** Ein Textblock stellt Zeichenketten formatiert dar. Zusätzlich können andere Typen von Medien darin integriert sein, wie z. B. Abbildungen.

- **T\_A1 Bezeichnung:** Ein Textblock besitzt eine Bezeichnung bzw. ein Titel (Coursera Inc., 2016).
- **T\_A2 Inhalt:** Der Inhalt eines Textblocks ist vornehmlich Zeichenketten, die formatiert dargestellt werden. Sie können außerdem Abbildungen, Uniform Resource Locator (URL) oder spezielle Formatierungen, wie beispielsweise Quellcode enthalten (z. B. Capernaum Ltd. (2016)).
- **T\_A3 Frage stellen:** Während dem Lesen eines Textes soll die Möglichkeit bestehen, eine Frage zu einem Text zu stellen und damit eine Diskussion zu starten (Fuchs, 2016).
- **T\_A4 Freigeschaltet:** Textblöcke sollen freigeschaltet<sup>40</sup> werden können (Ryzac, 2016; Khan Academy Inc., 2016).
- **T\_Z1 Einbettung in Aufgabe:** Textblöcke können in Aufgaben eingebettet sein, beispielsweise als Teil einer Fragestellung (Ryzac, 2016).

#### 4.4 ZWISCHENFAZIT

Dieses Kapitel stellte bereits durchgeführte Literature Reviews und deren Forschungsfokus vor. Es zeigt, dass Forschungslücken existieren, die durch eigene eigens durchgeführte Dokumentenanalyse teilweise geschlossen werden konnten. Für die Durchführung wurden die notwendigen theoretischen Grundlagen geschaffen sowie der Prozess und das letztendliche Vorgehen präsentiert. Die Ergebnisse der Dokumentenanalyse wurden systematisch aufbereitet und GDE erfasst. Aus Gründen der Verfestigung dieser Ergebnisse und durch einen subjektiv noch nicht abgeschlossenen Erkenntnisgewinn wurde eine darauffolgende Lernplattformanalyse durchgeführt. Diese lieferte bezogen auf die bereits erfassten GDE weitere wertvolle Erkenntnisse und konnte zudem darüber hinaus Konzepte offen legen, welche systematisch durch ein zuvor entworfenes Analyseschema beschrieben, kategorisiert und systematisch ausgewertet werden konnten.

<sup>40</sup> Unter einer Freischaltung wird in diesem Kontext auch die Erklärung zur Teilnahme an einer Lerneinheit verstanden.

Tabelle 15: Übersicht über identifizierte Elemente aus Dokumentenanalyse und Lernplattformanalyse

Element	Kürzel	Charakteristika	Typ
Badge	B_A1	Bildliche Repräsentation	Attribut
	B_A2	Titel	Attribut
	B_A3	Sharing	Attribut
	B_A4	Feedback	Attribut
	B_A5	Freigeschaltet	Attribut
	B_Z1	Transparenz	Zusammenhang
	B_Z2	Badges in Leaderboards	Zusammenhang
	B_Z3	Badges abhängig von Skill	Zusammenhang
	B_Z4	Badges abhängig von Level	Zusammenhang
	B_Z5	Badges abhängig von Badges	Zusammenhang
	B_Z6	Badges abhängig von Points	Zusammenhang
	B_R1	Vergabe als Belohnung bei Lernaktivitäten	Regel
Leaderboard	LB_A1	Sortierung	Attribut
	LB_A2	Repräsentationsformen	Attribut
	LB_A3	Reichweite	Attribut
	LB_Z1	Leaderboard abhängig von Achievements	Zusammenhang
Points	P_A1	System	Attribut
	P_A2	Feedback	Attribut
	P_A3	Sharing	Attribut
	P_A4	Repräsentation	Attribut
	P_A5	Veränderung	Attribut
	P_Z1	Points abhängig von Level	Zusammenhang
	P_Z2	Points durch Hilfe	Zusammenhang
	P_Z3	Points abhängig von Schwierigkeitsgrad	Zusammenhang
	P_Z4	Points abhängig zu Profil	Zusammenhang

weiter auf nächster Seite

Element	Kürzel	Charakteristika	Typ
	P_Z5	Points abhängig von Abschlussassessment	Zusammenhang
	P_R1	Belohnung mit Points	Regel
Level	L_A1	Progressives Freischalten	Attribut
	L_A2	Repräsentationsformen	Attribut
	L_A3	Feedback	Attribut
	L_A4	System	Attribut
	L_Z1	Levelaufstieg abhängig von Points	Zusammenhang
	L_Z2	Unlock abhängig von Level	Zusammenhang
	L_Z3	Levelaufstieg abhängig von Qualitygate	Zusammenhang
Quest	Q_A1	Kursüberblick	Attribut
	Q_A2	Instruktion	Attribut
	Q_A3	Feedback	Attribut
	Q_A4	Freiwilligkeit	Attribut
	Q_A5	Mehrfache Bearbeitung	Attribut
	Q_Z1	Quests als Teil von Erzählungen	Zusammenhang
	Q_Z2	Quests können vorgeschlagen werden	Zusammenhang
	Q_Z3	Feedback von Dozierenden	Zusammenhang
	Q_R1	Belohnung von Quests	Regel
Content Unlocking	CU_A1	Feedback	Attribut
	CU_A2	Transparenz	Attribut
	CU_Z1	Unlock abhängig von Points	Zusammenhang
	CU_Z2	Unlock abhängig von Lerneinheiten	Zusammenhang
Skill	S_A1	Freigeschaltet	Attribut
	S_A2	Feedback	Attribut
weiter auf nächster Seite			

Element	Kürzel	Charakteristika	Typ
	S_Z1	Skill abhängig von Lerneinheiten	Zusammenhang
Profil	PR_A1	Persönliche Daten	Attribut
	PR_A2	Zustand Lerneinheiten	Attribut
	PR_Z1	Profil abhängig von eingesetzten GDE	Zusammenhang
	PR_R1	Profilaktualisierung	Regel
Diskussion	D_A1	Anzahl Diskussionen	Attribut
	D_A2	Anzahl Kommentare	Attribut
	D_A3	Metadaten	Attribut
	D_A4	Kommentare	Attribut
	D_A5	Kommentare bewerten	Attribut
	D_A6	Suche	Attribut
	D_Z1	Diskussion zu Lernbaustein	Zusammenhang
Lerneinheit	LE_A1	Lernziele	Attribut
	LE_A2	Bezeichnung	Attribut
	LE_A3	Zeitaufwand	Attribut
Aufgabe	A_A1	Titel	Attribut
	A_A2	Fragentext	Attribut
	A_A3	Antworten	Attribut
	A_A4	Feedback	Attribut
	A_A5	Freigeschaltet	Attribut
	A_A6	Lesezeichen	Attribut
	A_A7	Anzahl Versuche	Attribut
	A_A8	Hilfestellungen	Attribut
	A_A9	Frage stellen	Attribut
	A_Z1	Aufgabe zu GDE	Zusammenhang
Video	V_A1	Bezeichnung	Attribut
	V_A2	Videoplayer	Attribut
	V_A3	Frage stellen	Attribut
	V_A4	Freigeschaltet	Attribut

weiter auf nächster Seite

Element	Kürzel	Charakteristika	Typ
	V_Z1	Einbettung in Aufgabe	Zusammenhang
	V_Z2	Video zu GDE	Zusammenhang
Podcast	PC_A1	Bezeichnung	Attribut
	PC_A2	Podcastplayer	Attribut
	PC_A3	Frage stellen	Attribut
	PC_A4	Freigeschaltet	Attribut
	PC_Z1	Einbettung in Aufgabe	Zusammenhang
	PC_Z2	Podcast zu GDE	Zusammenhang
Textblock	T_A1	Bezeichnung	Attribut
	T_A2	Inhalt	Attribut
	T_A3	Frage stellen	Attribut
	T_A4	Freigeschaltet	Attribut
	T_Z1	Einbettung in Aufgabe	Zusammenhang

Die gewonnenen Erkenntnisse aus der Dokumentenanalyse und der anschließend durchgeführten Lernplattformanalyse fungieren als konkrete Anforderungen für eine Systematisierung dieser Domäne. Hierfür fasst die Tabelle 15 alle identifizierten Beschreibungscharakteristika (n=91) zu den erfassten Elementen (n=14) in Kurzform zusammen. In einem nächsten Schritt sollen die gewonnenen Erkenntnisse in ein abstraktes Modell überführt werden, um die Konzepte gemäß dem Ansatz des DSM abbilden und weiter entwickeln zu können. Die für das Verständnis notwendigen Grundlagen dieser Art des Modellierungsansatzes werden zu Beginn des darauffolgenden Kapitels geschaffen. Diese Art des Vorgehens wurde zudem in Bartel et al. (2017b) veröffentlicht..

*„Design is where science and art break even.“*

— Robin Mathew

### 5.1 HINFÜHRUNG

Trotz der Beliebtheit und Verbreitung aktueller Lernplattformen, die spielifizierte Lerneinheiten anbieten, finden sich in der Literatur – wie bereits ausführlich aufgezeigt – diverse Aussagen, welche die Notwendigkeit von ausgefeilterer Funktionalität betonen, um spielifizierte Lernkonzepte erfolgreich abzubilden. So sprechen sich beispielsweise Dicheva u. Dichev für einen neuen Typus von Lernsoftware aus, welcher „[...] ein intelligentes Mentoring von formalem und spielifizierten Unterricht sowie informellen Lernendengruppen“ (Dicheva u. Dichev, 2016, S. 6) gleichermaßen unterstützt. Eine Lernplattform mit einer solchen Eigenschaft würde das Potential bergen, sowohl für Lehrende als auch für Lernende die Kontrolle über Lerninhalte und Lernsettings zu ermöglichen (Dicheva u. Dichev, 2016, S. 6). Lehrende wären ohne technologisch-fundierte Wissen, lediglich mit vertrauten Domänenkonzepten in der Lage, spielifizierte Lehr-Lernarrangements zu erstellen, welche von Lernenden den individuellen Vorlieben entsprechend genutzt werden können. Dabei fungiert die technologische Basis einer DSM-basierten Lösung als eine Art Wegbereiter und methodisches Korsett, das im Rahmen von definierten Regeln eine systematische Verbindung zwischen einer Idee für ein spielifiziertes Lernkonzept von Lehrenden bis hin zu einer auf individuelle Bedürfnisse eingehende Lernplattform für Lernende herstellt. Seitens der Lehrenden ermöglicht eine DSML, deren Notation sich an bekannten Domänenkonzepten orientiert, bereits umgesetzte und erfolgversprechende spielifizierte Lernkonzepte wiederzuverwenden und ggf. zu erweitern. Die Weitergabe von modellierten spielifizierten Lernkonzepten wird dadurch ermöglicht, dass die verwendete technologische Basis eine vergleichsweise geringe Kopplung an ein-

schlägige Inhalte aufweist. Durch diese Form der Abstraktion wird vermieden, dass umfangreiche Anpassungen<sup>1</sup> notwendig sind, um ein spielifiziertes Lernkonzept in einer Lernplattform abzubilden, wie es bei bestehenden Lernplattformen teilweise der Fall ist. Sollten erforderliche Anpassungen von bestehenden Plattformen nicht möglich sein, so kann das Eingehen von ungewollten Kompromissen bzgl. der Umsetzbarkeit eines spielifizierten Lernkonzepts dessen Potential erheblich mindern und schließlich zu einseitigen motivationalen Lernerfahrungen führen (Werbach, 2014, vgl. S. 271).

## 5.2 GRUNDLAGEN EINER DOMAIN-SPECIFIC MODELING LANGUAGE

Im Folgenden werden die Grundlagen von Sprachen sowie (domänenspezifischen) Modellierungssprachen beschrieben.

### 5.2.1 *Allgemeine Merkmale von Sprachen*

Aus einer logisch-philosophischen Sichtweise bilden Sprachen die Grenzen zu unserer Welt, indem sie festlegen, was beschrieben werden kann und was umschrieben werden muss (Wittgenstein, 2001). Sie sind nach Chomsky linguistisch zu definieren als eine festgelegte Sammlung von Sätzen mit einer endlichen Länge, die alle aus einem endlichen Alphabet von bedeutungsverleihenden Symbolen konstruiert werden (Chomsky, 1959, S. 137). Eine Sprache selbst ist jedoch eine nicht abgeschlossene Menge (Chomsky, 1959, S. 137). Aus dieser Definition von Chomsky und wie bereits vorher von Morris (1938) festgehalten lassen sich zunächst die Ebenen Syntax und Semantik<sup>2</sup> identifizieren, welche sich auch in Modellierungssprachen wiederfinden. Die Syntax einer Sprache spezifiziert die konzeptuelle Struktur, also deren Konstrukte, Eigenschaften und Verbindungen zueinander, während die Semantik den Sprachkonzepten eine Bedeutung verleiht (Kelly u. Tolvanen, 2008, S. 68f.). Bei einer natürlichen Sprache wie Englisch oder Deutsch findet sich ein hohes Maß an semantischer Mehrdeutigkeit für ein und dasselbe Sprachkonstrukt, sogar in dem selben Kontext (González Pérez u. Henderson-Sellers, 2008, S. 12). Außerdem können natürliche Sprachen als linear charakterisiert werden. Dies bedeutet, dass nicht-lineare Konstrukte (z.B. eine Tabelle) linear verarbeitet werden müssen, um sie zu kommunizieren (González Pérez u. Henderson-Sellers, 2008, S. 12).

<sup>1</sup> Im Speziellen betrifft dies nicht die Datenebene, sondern viel mehr die Konzeptebene.

<sup>2</sup> Morris (1938) nennt in diesem Zusammenhang noch die Pragmatik als Grundlage seiner Zeichenlehre.



### 5.2.2 Modellierungssprachen

Im Gegensatz zu natürlichen Sprachen lassen Modellierungssprachen derartige Variationen nicht zu. Sie besitzen eine höhere Stringenz, welche in einem Korsett aus Regeln eingebettet und idealerweise frei von Redundanzen sind (Figas et al., 2015a, S. 650). Die UML beispielsweise repräsentiert einen Standard<sup>3</sup> im Software Engineering, ist Teil des Model Driven Architecture (MDA)<sup>4</sup>-Ansatzes der OMG und stellt eine weitere Abstraktion oberhalb von gängigen Programmierhochsprachen (z.B. Java oder C++) dar (Soley, 2000, S. 4). Sie ist eine generische objektorientierte Modellierungssprache und ist in der Lage sich selbst durch Klassendiagramme zu beschreiben. Das ist eine rekursive Sicht, die es erlaubt die UML dafür zu nutzen, fast alles durch sie (z.B. in Form einer Klasse) zu beschreiben (González Pérez u. Henderson-Sellers, 2008, S. 13). Dieser Umstand macht diese Modellierungssprache zwar mächtig, aber erfordert gleichzeitig auch ein hohes Maß an Abstraktion, denn bei einer Beschreibung eines (realen) Objekts durch eine Klasse, gehen Informationen zu diesem Objekt verloren. Sollte wie für die UML intendiert mit Hilfe von Ansätzen aus dem MDSD plattformspezifischer Code aus Modellen generiert werden, sind die Möglichkeiten dieses *one-size-fits-all*-Ansatzes begrenzt. Die Interoperabilität und Portabilität sind also eingeschränkt, da die UML nicht über die Codeebene hinaus abstrahiert und semantische Zuordnungen von Modellbestandteilen zu Quellcode sogar innerhalb eines Teams voneinander abweichend sein können (Kelly u. Tolvanen, 2008, S. 55f.). Die Sprachkonstrukte und Konzepte der jeweiligen Programmiersprache sind teilweise so verschieden, dass dieser Ansatz nicht mehr tragfähig ist, ohne einen erhöhten Aufwand in die Zuordnung von UML- zu Programmier-Sprachkonstrukt (z.B. mit Hilfe der Object Constraint Language (OCL)) zu investieren und anschließend die fehlenden Informationen, die durch das Modellieren entstanden sind, im Code manuell zu ergänzen (Stahl u. Völter, 2006, S. 12). Dabei handelt es sich vornehmlich um Informationen oder Regeln, die bedingt durch den Beschreibungskontext des Modells relevant sind. Folglich ist der Anteil an Quellcode sehr klein, der direkt aus den Diagrammen der UML generiert werden kann, verglichen mit der Gesamtmenge an Quellcode einer Software (Kelly u. Tolvanen, 2008, S. 56). Auch wenn diese Form des Informationsverlustes bei allen Modellierungstä-

<sup>3</sup> In zweifacher Hinsicht: Sowohl durch ihre Verbreitung in der Disziplin, als auch durch eine offizielle Standardisierung als ISO-Norm (OMG, 2011).

<sup>4</sup> Nach González Pérez u. Henderson-Sellers sind Model Driven Engineering (MDE) und Model Driven Development (MDD) als weitere sehr eng verwandte Akronyme zueinander zu betrachten, wobei es sich bei MDE um den am meisten generischen Begriff für das Object Management Group (OMG)-spezifische Konzept MDA handelt (González Pérez u. Henderson-Sellers, 2008, S. 65). MDD ist dabei ein Synonym für Model Driven Software Design (MDSD).

tigkeiten vorzufinden ist, gibt es Typen von Modellierungssprachen, die für diesen Verlust weniger anfällig sind. Dies liegt unter anderem daran, dass sie für einen bestimmten Zweck innerhalb eines definierten Kontexts erstellt werden. Als Teil des MDSD-Ansatzes sind sie weniger expressiv, dadurch aber spezifischer in der Anwendung (Fowler, 2010, S. 27).

### 5.2.3 Domänenspezifische Modellierungssprachen

Bei diesen Typen von Sprachen, den sogenannten domänenspezifischen Modellierungssprachen (engl.: DSML<sup>5</sup>), liegt eine Domäne zu Grunde, für die die Sprachen definiert werden, um sie dort möglichst effektiv nutzen zu können. Hierzu wird das Abstraktionslevel angehoben, um Lösungen zu Problemen zu erstellen, die direkt Konzepte und Regeln der spezifischen Problem-domäne nutzen und sie in eine Lösungsdomäne übertragen (siehe Abbildung 29<sup>6</sup>) (Kelly u. Tolvanen, 2008, S. 15). So wird eine zusätzliche Flexibilität gewonnen, indem die Problemlösung zunächst in Form eines domänenspezifischen Modells vorliegt, dieses mit Hilfe eines oder mehrerer Generatoren konkretem Quellcode zugeordnet wird, was schließlich zu ausführbarem Code führt (Kelly u. Tolvanen, 2008, S. 15). Abbildung 30 veranschaulicht diesen Zusammenhang.

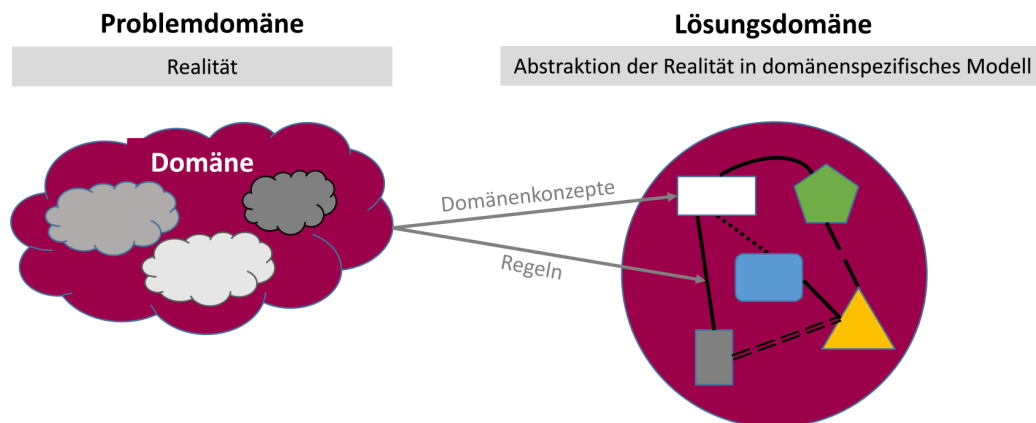


Abbildung 29: Problem- und Lösungsdomäne im Kontext von DSM

<sup>5</sup> Dieser Begriff existiert um eine vornehmlich grafische oder textuelle Sprache, also das Werkzeug von DSM zu betiteln. Der Begriff DSM hingegen, wurde geprägt von Juha-Pekka Tolvanen und beschreibt das dahinter liegende Prinzip (Kelly u. Tolvanen, 2008). Im Unterschied zu einer DSML wird eine Domain-Specific Language (DSL) hauptsächlich mit textuellen domänenspezifischen Sprachen in der Literatur assoziiert auch wenn die Akronyme für grafische und textuelle Sprachen häufig als Synonym verwendet werden (Stahl u. Völter, 2006; Fowler, 2010).

<sup>6</sup> In ähnlicher Darstellung bereits veröffentlicht in Bartel et al. (2017b, S. 1502).

Dies führt dazu, dass Entwicklungsschritte, die im „klassischen“ Software Engineering notwendig sind, wie beispielsweise die Architektur einer Software, in den Hintergrund rücken und man sich stattdessen auf die problemlösende Benutzung einer domänenspezifischen Modellierungssprache konzentrieren kann. Damit ist deren Nutzung nicht mehr nur für Personen möglich, die mit den technischen Details vertraut sind, sondern auch für jene geeignet, die sich als Domänenexperten bezeichnen würden (Kelly u. Tolvanen, 2008, S. 22). Sie erstellen Modelle mit aus der Domäne vertrauten Begriffen und Regeln, die direkt weiter verarbeitet werden können. Außerdem verstecken DSMLs Komplexität und unterstützen „technische Entwickler, Programmierer und weniger erfahrene Entwickler dabei, effektiv Software zu entwickeln“ (Kelly u. Tolvanen, 2008, S. 31, eigene Übersetzung). Aus diesem Grund ist die Entwicklungszeit von Applikationen, die mit einer DSML erstellt werden, wesentlich kürzer, im Gegensatz zu generischen Sprachen wie der UML, bei der das Modell manuell verarbeitet werden muss (siehe Abbildung 30) (Kelly u. Tolvanen, 2008, S. 14) (Stahl u. Völter, 2006, S. 13). Der Zuwachs an Produktivität wird dabei auf einen Bereich zwischen 300-1000 Prozent<sup>7</sup> geschätzt, verglichen mit einem Einsatz von generischen Modellierungssprachen oder manuellem Programmieren (Kelly u. Tolvanen, 2000; Tolvanen u. Kelly, 2016; Weiss u. Lai, 1999).

Diesem Vorteil gegenübergestellt werden muss der Aufwand, der für die vollständige Entwicklung einer adäquaten DSML notwendig ist (z. B. López-Fernández et al. (2016, S. 101f.)). Dieser wird im unternehmerischen Bereich anfangs als höher eingeschätzt im Vergleich zu einem Einsatz von generischen Modellierungssprachen, sinkt jedoch schon nach kurzer Zeit durch das stetige Lernen der Anwendung der Sprache durch deren Anwender (Kelly u. Tolvanen, 2008, S. 32).

Weiterhin können DSML dazu beitragen, die Softwarequalität zu steigern, da ein erhöhtes Maß an Automation bei der Transformation von derartigen Modellen in Quellcode auftritt (Kelly u. Tolvanen, 2008, S. 14) (Stahl u. Völter, 2006, S. 13). Diese bedienen sich dabei Korrektheitsregeln, die direkt aus der Domäne stammen und es erschweren, Modelle mit schlechten oder unzu-

---

<sup>7</sup> Kelly u. Tolvanen berichten beispielsweise über einen Produktivitätszuwachs bei der Applikationserstellung für Nokia von 1000 Prozent zu vorherigen manuellen Praktiken (Kelly u. Tolvanen, 2000; Tolvanen u. Kelly, 2016). Vermutlich aus Datenschutzgründen ist jedoch unklar wie diese manuellen Praktiken vor der Einführung von DSM konkret ausgestaltet waren und an welchen Punkten Veränderungen vorgenommen wurden. Ebenso kann nicht bewertet werden, welche bedarfsspezifischen Anpassungen an der DSML nach Einführung vorgenommen werden mussten und wie diese in Sachen Produktivität zu bewerten waren.

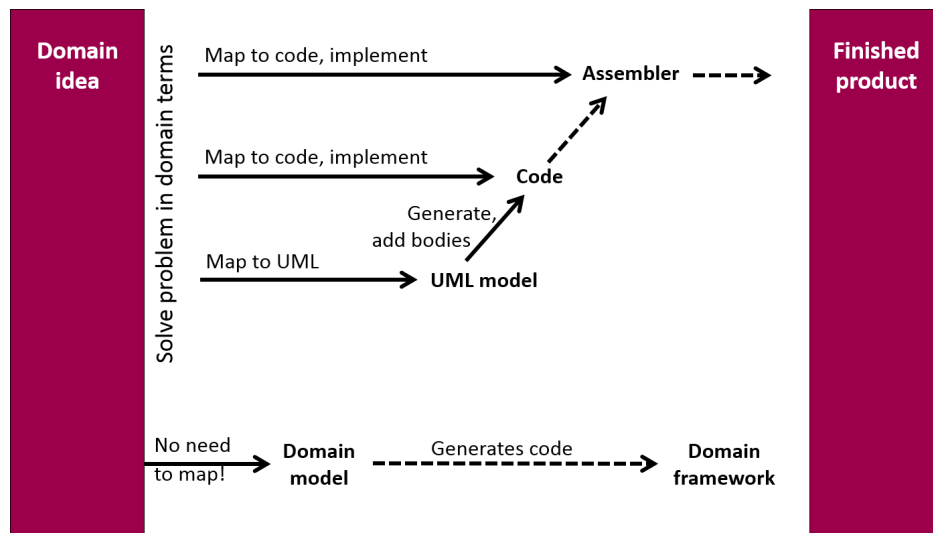


Abbildung 30: Übergang von Domänenidee bis zum fertigen Produkt - Traditionelle Softwareentwicklung und DSM im Vergleich (Kelly u. Tolvanen, 2008, S. 16)

lässigen Spezifikationen zu erstellen (Kelly u. Tolvanen, 2008, S. 27). Kelly u. Tolvanen formulieren diese These unterstützend:

„By automating the mapping from a problem domain to a solution domain we also reduce the risk of the implementation not corresponding to the solution specified in the problem domain“ (Kelly u. Tolvanen, 2008, S. 27).

### 5.2.3.1 Abgrenzung zu MDSD

Bei DSM wird also darauf abgezielt, direkt aus einem oder mehreren vorhandenen Modellen automatisiert Code zu generieren. Dabei wird DSM als eine Instanz von MDSD gesehen (Tolvanen u. Kelly, 2016). Das Ergebnis von MDSD ist zwar dasselbe, jedoch unterscheidet der Ansatz sich von DSM vor allem darin, dass „model transformations normally mean that during each step developers extend the automatically produced models with further details“ (Kelly u. Tolvanen, 2008, S. 57).

Ebenso entfällt beim DSM ein manuelles Editieren des aus dem Modell generierten Codes, was weitere Fehler im Vorfeld vermeiden kann und damit beispielsweise in einem unternehmerischen Umfeld Kosten für die Fehlerkorrektur einsparen kann (Tolvanen, 2005, S. 18). Dies setzt jedoch eine entsprechende Korrektheit der Generatoren voraus, die für die Codegenerierung zuständig sind. Werden dennoch Fehler in dem generierten Code gefunden, können diese bedingt durch das in der Verarbeitungsstruktur vorherrschenden

Separation of Concerns (SoC)-Prinzip<sup>8</sup> (Dijkstra, 1982, S. 60ff.) einfacher behoben werden, was langfristig eine bessere Wartbarkeit und ein transparentes Änderungsmanagement ermöglichen (Stahl u. Völter, 2006, S. 13).

#### 5.2.3.2 *Erweiter- und Anpassbarkeit*

Sollten während des Einsatzes einer DSML dennoch innerhalb der Domäne valide Spezifikationen auftauchen, die bei der Erstellung der Sprache nicht berücksichtigt wurden, muss die Sprache entsprechend erweitert werden (einmal definierte Spezifikationen lassen sich jedoch für unterschiedliche Problemlösungen wiederverwenden). Für die UML stellt eine derartige Erweiterung jedoch eine Herausforderung dar, weshalb UML-Profile (Kecher u. Salvanos, 2015, S. 435ff.) Abhilfe schaffen können. UML-Profile erweitern die allgemeine UML um generische Einschränkungen einer Domäne, also weitere Elemente, Stereotypen oder domänenspezifische Einschränkungen mit OCL (Kelly u. Tolvanen, 2008, S. 58). Ein bekanntes und verbreitetes Beispiel eines Profils stellt die Systems Modeling Language (SysML) dar (Burger, 2014, S. 250f.). Jedoch sind UML-Profile nur für Einsatzzwecke gedacht, die sehr nah an der generischen UML liegen, da sonst das Mapping von generischer Sprache zu spezifischen Konzept mit ähnlichem Aufwand, wie bereits oben für die UML beschrieben, verbunden wäre (Kelly u. Tolvanen, 2008, S. 58).

Beim DSM können also nur die Konzepte instanziiert werden, die bei der Spracherstellung berücksichtigt wurden (Gabrysiak et al., 2011). In diesem Zusammenhang spricht Fowler von einer limitierten Expressivität, die jedoch mit einer leichteren Erlernbarkeit, bedingt durch den Domänenfokus, im Vergleich zu einer generischen Sprache einhergeht (Fowler, 2010, S. 27) und formuliert: „The domain focus is what makes a limited language worthwhile“ (Fowler, 2010, S. 28). Dieser Umstand erlaubt ebenso eine starke und vor allem frühe Einbindung von Domänenexperten bei der Erstellung einer solchen Sprache. In dem Fall fungiert eine DSML nicht nur als das zu erstellende und problemlösende Artefakt für eine spezifische Domäne, sondern vielmehr auch als eine einheitliche Terminologie, was für eine Vermeidung von Missverständnissen in der Kommunikation unabdingbar ist (Rupp, 2009).

Die Abbildung 31 vergleicht domänenspezifische Modellierungssprachen mit weiteren Sprachtypen im Hinblick auf die Anzahl an Sprachelementen, Abstraktionsgrad der Sprache und die Anzahl verschiedener Anwendungsgebiete (dargestellt durch die Größe der einzelnen Kreise). Durch einen relativen

<sup>8</sup> In diesem Fall bezieht sich das SoC-Architekturprinzip auf die strukturelle Unterteilung von Modell und Generator sowie die jeweils dazugehörigen Subeinheiten, welche jeweils nur für einen einzelnen Aspekt verantwortlich sind.

Vergleich wird deutlich, wie stark die limitierte Expressivität im Vergleich zu generischen Modellierungssprachen ist bei gleichzeitig kleiner Anzahl an Elementen einer DSML.

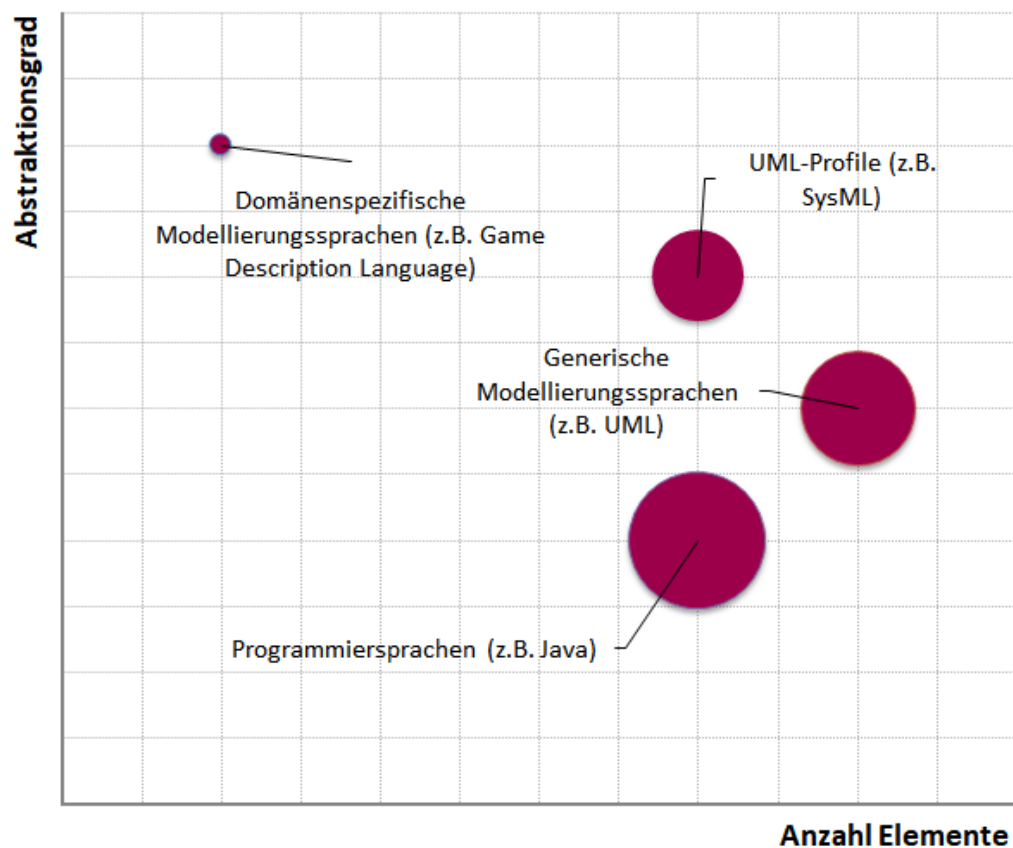


Abbildung 31: Relativer Vergleich domänenspezifische Modellierungssprachen mit weiteren Sprachtypen (auf Basis von Kleppe (2009, S. 32))

### 5.3 BESTANDTEILE VON DOMAIN-SPECIFIC MODELING

Im Folgenden soll näher auf die Bestandteile von DSM eingegangen werden, um ein besseres Verständnis dafür zu bekommen, welche Aspekte bei der Erstellung einer derartigen Sprache Berücksichtigung finden sollen und wie sich eine DSML in das Konzept von DSM eingliedern lässt. Nach Kelly u. Tolvanen handelt es sich bei den Bestandteilen, die DSM ausmachen, um die Sprache (DSML) bzw. dazugehörige Modelle, Generatoren, welche die Modelle umsetzen, und das Domain Framework, in welches der generierte Code eingebettet wird (Kelly u. Tolvanen, 2008, S. 68, S. 341).

### 5.3.1 *Sprache*

Wie bereits angedeutet, lassen sich die Bestandteile Syntax (Modellierungskonzepte, ihre Eigenschaften und Verbindungen zueinander (Kelly u. Tolvanen, 2008, S. 68)) und Semantik (Bedeutung der Modellierungskonzepte) bei Sprachen identifizieren. Eine DSML lässt eine weitere Unterteilung in die zwei Typen abstrakte und konkrete Syntax zu (Kelly u. Tolvanen, 2008, S. 68; Clark et al., 2004).

#### 5.3.1.1 *Konkrete Syntax*

Bei einer konkreten Syntax handelt es sich um einen Formalismus in Form einer Notation, welcher eine visuelle Schnittstelle für den Anwender einer DSML herstellt (Kelly u. Tolvanen, 2008, S. 70; Stahl u. Völter, 2006, S. 57). Kelly u. Tolvanen geben an, dass die Konstrukte eine Kombination aus grafischen (Formen und Symbole) und textuellen (wohldefinierte Kontrollstrukturen, für Text oder weitere Kontrollstrukturen) Elementen darstellen, welche durch den Nutzer zur Modellierung kombiniert und parametrisiert werden können (Kelly u. Tolvanen, 2008, S. 70). Dabei können Text, Grafiken, Tabellen oder Matrizen gängige Möglichkeiten der Darstellung sein (Völter et al., 2013, S. 26; Kelly u. Tolvanen, 2008, S. 70; Stahl u. Völter, 2006, S. 57). Ebenfalls sind Kombinationen der Darstellungstypen möglich (Völter et al., 2013, S. 26; Millett u. Tune, 2015). Dabei sollte für jedes Domänenkonzept genau eine nicht redundante und adäquate Darstellungsform gewählt werden. Dies bedeutet, dass der Wiedererkennungswert für Anwender der Sprache dadurch gesteigert werden soll, indem die visuelle Notation an das Domänenkonzept angelehnt ist und jede Notation nur genau ein Domänenkonzept repräsentiert (Kelly u. Tolvanen, 2008, S. 70).

Ein Beispiel für eine textuelle und grafische Repräsentation stellt Abbildung 32 dar. Der obere Teil der Abbildung widmet sich der textuellen<sup>9</sup> Darstellung einer konkreten Syntax, welche zur rollenbasierten Rechtevergabe für die Ausführung von Operationen innerhalb eines Systems entwickelt wurde (Strembeck u. Zdun, 2009, S. 1276). Der untere Teil veranschaulicht eine DSML grafisch<sup>10</sup>, mit der Benutzerschnittstellenkonzepte eines Auto-Infotainmentsystems dargestellt werden können. Letztere erlaubt nicht nur die Logik eines solchen Systems mit domänenspezifischen Konzepten (Buttons, Displays und Menüs) zu nutzen, sondern auch dynamische Abfolgen zwischen den Elemen-

<sup>9</sup> Auszug aus Strembeck u. Zdun (2009, S. 1281).

<sup>10</sup> Siehe <http://www.metacase.com/cases/autoinfo.html>, abgerufen am 26.01.2016

ten grafisch zu erstellen, aus denen plattformspezifischer und auf dem Infotainmentsystem lauffähiger Code generiert werden kann.

#### Rollenbasierte Zugangskontrolle – textuelle DSML

```

Role create Analyst
Role create JuniorAnalyst
Analyst addJuniorRole JuniorAnalyst
Role create InternalRevision
Role create Trader
...
Permission create Read_Report -operation read -object report
Permission create Write_Report -operation write -object report
Permission create Edit_Report -operation edit -object report
...
JuniorAnalyst assignPermission Read_Report
JuniorAnalyst assignPermission Edit_Report
...
Subject create Sarah
Sarah assignRole Analyst
Sarah assignRole Trader

Subject create Sophie
Sophie assignRole InternalRevision
...

```

#### Car Infotainment - grafische DSML

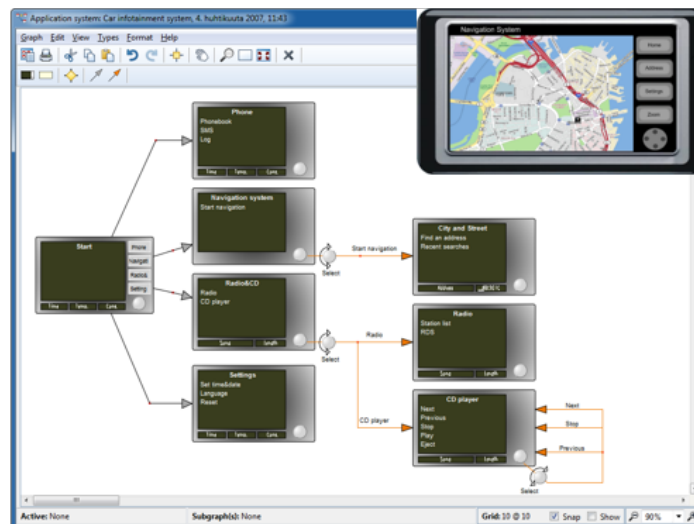


Abbildung 32: Beispiele für eine textuelle und grafische konkrete Syntax

##### 5.3.1.2 Abstrakte Syntax und Metamodelle

Die abstrakte Syntax spezifiziert die (Daten-) Struktur der semantisch relevanten Informationen einer DSML und deren grammatikalischen Regeln zur Verknüpfung (Völter et al., 2013, S. 26; Kelly u. Tolvanen, 2008, S. 68). Die abstrakte Syntax einer DSML wird in einem Metamodell basierend auf einer Domäne definiert. Die konkrete Syntax ist also eine Realisierung (Mapping (Clark et al., 2004)) der abstrakten Syntax, wobei es zu einer abstrakten Syn-



tax mehrere Ausprägungen von konkreten Syntaxen (entspricht Notationen) geben kann (Stahl u. Völter, 2006, S. 57; Kleppe, 2009, S. 75f.). Dies wird deutlich, wenn man sich ein Beispiel vor Augen führt: Die Sätze *Bob bietet Alice ein Eis an* und *Alice wird ein Eis von Bob angeboten* repräsentieren die gleiche Aussage. Jedoch ist diese in beiden Sätzen unterschiedlich repräsentiert, d.h. beide Sätze besitzen eine unterschiedliche Notation. Die abstrakte Syntax beider Sätze ist jedoch identisch und definiert die dahinter liegende, für den Leser nicht unmittelbar ersichtliche Satzstruktur, welche üblicherweise in Metamodellen festgehalten wird.

Folgt man der Definition von Clark et al., handelt es sich bei einem Metamodell allgemein um ein Modell einer Modellierungssprache, welches die Modellierungssprache auf einer abstrakteren Ebene beschreibt, als sie selbst es tut (Clark et al., 2004, S. 19). Aus objektorientierter Sicht, könnte man also von einer Klasse-Objekt-Beziehung sprechen, bei der die Klasse alle Eigenschaften und Beziehungen allgemein beschreibt und ein Modell ein konkretes Objekt bzw. Instanz dieser Klasse bildet (González Pérez u. Henderson-Sellers, 2008, S. 17). In der Literatur sind weitere Eigenschaften zu finden, wie sich Metamodelle von Modellen abgrenzen lassen, und qualitative<sup>11</sup> Kriterien, die für Metamodelle gelten sollen (Clark et al., 2004, S. 19; González Pérez u. Henderson-Sellers, 2008, S. 17; Flatscher, 2002, S. 324; Atkinson, 1999, S. 24; Kelly u. Tolvanen, 2008, S. 76; Bertoa u. Vallecillo, 2010, S. 4ff.; Seidewitz, 2003, S. 26ff.):

- Ein Metamodell muss in der Lage sein, die abstrakte Syntax einer Sprache sowie deren Semantik zu beschreiben.
- Ein Metamodell soll für die Erstellung eines Modells anleiten (z. B. durch Verwendung einschlägiger und prägnanter Begriffe aus der Domäne, ohne sprachliche Veränderungen, wie Tilgung, Verzerrung oder Generalisierung (Rupp, 2009, S. 120ff.; Bandler u. Grinder, 1994)), unnötige Details vor dem Benutzer verstecken, aber trotzdem alle relevanten Modellierungskonzepte der Domäne erfassen.
- Modellierungskonzepte einer Domäne sollen in einem Metamodell redundanzfrei und mit möglichst wenigen strukturellen Indirektionen (z. B.

<sup>11</sup> Hierbei ist zu beachten, dass eine Messbarkeit eines Kriteriums nicht immer objektiv erfolgen kann. Subjektive Kriterien, wie die Verständlichkeit, was für Metamodelle gelten soll, sind abhängig von einer entsprechenden Einschätzung durch den Benutzer. Ohnehin stellen derartige weiche Faktoren kein unmittelbares Qualitätsmerkmal nur für DSML dar, sondern besitzen ein höheres Maß an Allgemeingültigkeit.

unnötige Abstraktion durch zusätzliche Vererbung) beschrieben und verwendet werden und dabei unterstützen, aus den daraus erstellten Modellen Software generieren zu können.

- Ein Metamodell muss in einer Metamodellarchitektur eingebettet sein. Darunter verstehen Clark et al. eine Beschreibung eines Metamodells durch ein weiteres Modell, das Meta-Metamodell. Dies ist in der Lage, alle Instanzen eines Metamodells durch ein Meta-Metamodell und zudem sich und seine Elemente selbst durch Instanzen zu beschreiben<sup>12</sup>. Die Abbildung 33 illustriert diese Dualität an einem Beispiel für die UML. Darin wird die Beziehung zwischen Klassen und Attributen mit Hilfe eines Klassendiagramms beschrieben und gleichzeitig die innere Struktur des Konzepts Attribut (engl. Property) innerhalb einer Klasse definiert.

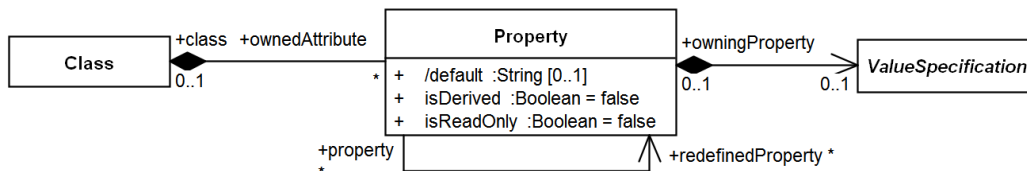


Abbildung 33: Beziehung zwischen Klasse und Attribut in einem Meta-Metamodell (Auszug aus OMG (2014, S. 42))

- Metamodelling, also die Wissenschaft und das Erstellen eines Metamodells und dessen Instanzen, ist die Art zu modellieren, bei der der Input (Metamodell) den selben Typ wie der Output (Modell) besitzt. Nach dieser These wäre also jede (generische) Modellierungssprache (z. B. die UML) für sich wieder ein Metamodell bezogen auf eine konkrete Anwendung dessen (z. B. um ein Kundenverwaltungssystem zu modellieren). Dabei ergibt sich eine rekursive Struktur, deren Tiefe nur durch die zunehmende Komplexität begrenzt wird. Ebenso ist eine begriffliche Unschärfe zu beobachten, die sich vor allem in der Anwendung einer Metamodellarchitektur zeigt.

Die Tabelle 16 fasst dieses hierarchische Modell beispielhaft für die UML zusammen, um die einzelnen Abstraktionsstufen der dabei entstehenden Metamodellarchitektur besser voneinander abgrenzen zu können. Es handelt sich dabei um das von der OMG als Standard entwickelte Meta Object Facility (MOF)<sup>13</sup> (OMG, 2014). Die oben erwähnte begriffliche Unschärfe rührt daher,

<sup>12</sup> Seidewitz führte für Meta-Metamodelle aufgrund ihrer Selbstbeschreibungsfähigkeit im Bezug auf Metamodelle den Begriff reflexives Metamodell ein (Seidewitz, 2003, S. 29).

<sup>13</sup> Für diese Arbeit soll das Akronym MOF verwendet werden, wohl wissend, dass ab UML 2.0 der Standard überarbeitet wurde und die neue Bezeichnung Meta Object Facility 2 (MOF2) trägt (OMG, 2014).

dass wie Atkinson beschreibt, der Begriff *meta* sowohl absolut (Bezeichnung einer Stufe) als auch relativ (Beziehung zur darüber- oder darunterliegenden Stufe) verwendet wird, um einzelne Abstraktionsstufen zu benennen (Atkinson, 1999, S. 23). Zur Vermeidung von Verwechslungen werden in dieser Arbeit alle Stufen über ihre Nummer  $M_0$  bis  $M_3$  bezeichnet. Eine weitere Klarstellung muss für die Bezeichnung des MOF als ein Meta-Metamodell erfolgen. Die OMG ist von dieser ursprünglichen Betitelung abgewichen und bezeichnet MOF nun, entgegen der weitläufigen Nennung, als ein Metamodell für die UML (OMG, 2011, S. 17).

Stufe	Repräsentation	Beispiel
$M_3$	Meta-Metamodell (MOF)	Das Modellierungskonzept <i>Klasse</i> besitzt (lokale) <i>Attribute</i> .
$M_2$	Metamodell (UML)	<i>Klasse</i> mit <i>Attribut</i> .
$M_1$	Model	Klasse <i>Person</i> mit Attribut <i>groesse</i> .
$M_0$	(Benutzer-)Daten	Objekt $p$ der Klasse <i>Person</i> Attributwert <i>190</i> des Attributs <i>groesse</i> .

Tabelle 16: Abstraktionsstufen einer MOF Metamodellarchitektur (angelehnt an OMG (2014) bzw. Atkinson (1999, S. 24f.))

Zwischen den einzelnen Abstraktionsstufen bestehen paarweise explizite *instance-of*-Beziehungen (Atkinson, 1999, S. 22; González Pérez u. Henderson-Sellers, 2008, S. 42). Präziser formuliert, muss jedes Element einer Stufe  $n$  genau eine Instanz eines Elements der Stufe  $n+1$  sein (OMG, 2011, S. 17). Von Stufe zu Stufe nimmt die Abstraktion ab, beginnend bei der Stufe mit der höchsten Abstraktion, der MOF ( $M_3$ ). Bei Letzterer handelt es sich um objektorientierte Modellierungskonzepte wie beispielsweise Klassen, Assoziationen oder Attribute, die für das Modellieren eines UML-Metamodells genutzt werden können (OMG, 2014, S. 27ff.). Kelly u. Tolvanen geben an, dass dieses Typ- und Instanzmapping, also die implizite Zuordnung von Elementen aus  $M_3$  zu deren Verwendung in  $M_2$  bei Menschen intuitiv geschieht, da man sonst nicht in der Lage wäre, darunterliegende Abstraktionsstufen auf Anhieb zu verstehen (Kelly u. Tolvanen, 2008, S. 76). Die darunterliegende Stufe  $M_2$  besteht aus einem oder mehreren Metamodellen, die mit den Modellierungskonzepten aus  $M_3$  definiert werden. Die Stufe  $M_1$  ist eine konkrete Klasse, die einen

Kontextbezug besitzt. Eine weitere Stufe tiefer, in Mo, finden sich schließlich die zustandsbehafteten<sup>14</sup> Objekte einer Klasse.

Dieses vierschichtige MOF Architekturmodell lässt sich auch auf weitere Anwendungsgebiete außerhalb der UML übertragen. Die Literatur zeigt eine Vielzahl an Anwendungsfeldern, jedoch wird angenommen, dass es mindestens dieser vier Schichten bedarf „to integrate the modeling of the usage and evolution of systems“ (Kelly u. Tolvanen, 2008, S. 75). Die oberste Schicht (M3) muss dabei immer das Prinzip der Selbstbeschreibung erfüllen (OMG, 2011, S. 18). Die Tabelle 17 zeigt beispielhaft zwei weitere bekannte Formalismen für Metamodelle sowie deren interne Repräsentationen.

Stufe	SQL	XML
M3	Relationales Modell und Tupellogik	XML Schema Sprache
M2	SQL-Syntax und Semantik	XML-Schema
M1	Elemente aus DDL, DML, DCL in Programmiersprachen, Dokumenten oder Routinen	XML-Dokumente
Mo	Abhängig von DBS und Einbettung z. B. gerichteter Graph (für DDL)	Baumstruktur (DOM/SAX) aus XML-Elementen

Tabelle 17: Beispielhafte Metamodell-Formalismen für SQL und XML

Diese Annahmen und das vierschichtige Architekturmodell treffen auch für den Bereich der (grafischen) DSML zu.

#### 5.3.1.3 Constraints bei der abstrakten Syntax

Der wesentliche Beitrag der abstrakten Syntax zu einer DSML ist die Definition von Regeln, die direkt aus der Domäne abgeleitet werden (Kelly u. Tolvanen, 2008, S. 69). Sie geben einen Rahmen dafür, wie Modelle, also konkrete Instanzen des Metamodells, umgesetzt werden können und begrenzen damit aktiv den durch die Domäne bereits begrenzten Freiraum bei der Erstellung von Modellen (Kelly u. Tolvanen, 2008, S. 69). Somit definieren sie gültige Werte, welche Beziehungen zwischen Modellierungskonzepten auftreten und wie bestimmte Modellierungskonzepte angewendet werden. Dafür kann, wie bereits erwähnt, eine eigens dafür entwickelte domänenspezifische Sprache

<sup>14</sup> Hier wird das Verständnis zu Grunde gelegt, dass unter dem Zustand eines Objekts zu einer Klasse, die Belegung der Attribute mit Attributwerten verstanden wird.

eingesetzt werden, die OCL (Combemale et al., 2017, S. 33). Kelly u. Tolvanen unterscheiden dabei zwei Typen von Regeln, die bei einer Erstellung einer DSML berücksichtigt werden sollen (Kelly u. Tolvanen, 2008, S. 69, S. 251f.):

- Korrektheits- und Konsistenzregeln: Die Korrektheits- und Konsistenzregeln zielen vor allem auf die syntaktische Fehlerfreiheit von Modellen ab. Diese können direkt aus der Domäne stammen und damit auch als Domänenregel fungieren.

**Beispiel:** In einem UML-Klassendiagrammen muss für jede Methode eine der Sichtbarkeiten *public*, *private*, *protected* oder *package* definiert werden.

- Domänenregeln: Bei den Domänenregeln handelt es sich - wie der Name bereits intendiert - um Regeln, die eine Domäne in einer Modellsicht widerspiegeln und damit die Grenzen der Domäne in einem Modell abbilden. Dabei können sie gültige Werte definieren, Beziehungen zwischen Modellierungskonzepten festlegen und wie bestimmte Konzepte im Modell zur Anwendung kommen.

**Beispiel:** In einer DSML, mit der es möglich ist, ein elektrisches Garagentor zu steuern, wäre es denkbar, die folgende Domänenregel vorzufinden:

*Solange der Motor läuft, welcher für das Öffnen und Schließen des Garagentors verantwortlich ist, muss eine rote Signallampe blinken.*

Aus dem Beispiel lässt sich folgern, dass die Domänenkonzepte Motor (-steuerung) und Signallampe eine Abhängigkeit (z. B. in Form einer wenn-dann-Beziehung) besitzen, die entsprechend bei der Modellierung einfließen muss. Fehlt dieser Teil und die Regel wird somit verletzt, muss der Benutzer entsprechend darüber in Kenntnis gesetzt werden, entweder als Ergebnis einer Überprüfung während des Modellierens oder spätestens bei dem Generieren von Quellcode durch einen Generator (Kelly u. Tolvanen, 2008, S. 252). Der Vorteil, den eine Domänenspezifität im Vergleich zur UML oder selbst zu UML-Profilen bietet, ist die Anwendbarkeit von Domänenregeln, die über schlichte Korrektheits- und Konsistenzregeln hinaus gehen und damit in erhöhtem Maße zur semantischen Korrektheit eines Modells beitragen können (Dalgarno u. Fowler, 2008, S. 5). Wie man sieht, könnte es sich bei dieser Regeltypologie auch um nichtfunktionale Anforderungen (Rupp, 2009) im Sinne einer Sicherheitsanforderung handeln, die sich direkt aus der Domäne ergeben und an das ausführende System gestellt werden. Folgern lässt sich daraus deren Wichtigkeit für die Erstellung einer DSML.

Für beide Typen von Regeln muss für jede einzelne konkrete Regel entschieden werden, wie damit bei einem Regelverstoß umgegangen wird. Regelverstöße können beispielsweise nur temporär auftreten, also während des Modellierens. Bezogen auf das oben genannte Beispiel eines UML-Klassendiagramms, in dem jede Methode mit einer der Sichtbarkeiten belegt werden muss, würde dies bedeuten, dass während oder nachdem der Methodename vom Benutzer eingegeben wird, die Sichtbarkeit noch nicht verfügbar sein muss. In diesem Moment befindet sich das Modell laut Regeldefinition in einem nicht konsistenten Zustand, der temporär ist, sofern der Benutzer die fehlende Sichtbarkeit noch ergänzt. Dieser Umstand hat also zur Folge, dass nicht alle Regeln nach einer atomaren Veränderung des Modells durch den Benutzer zur Überprüfung auf das Modell angewandt werden können, da deren Anwendung zu Fehlhinweisen führen würde, die für den Benutzer verwirrend sein können. Vielmehr müssen grundlegende Regeln definiert werden, die effizient überprüfbar sind und nicht anfällig sind für mögliche temporär inkonsistente Zustände des Modells. Ein Beispiel hierfür ist, eine Mehrfachvererbung innerhalb eines UML-Klassendiagramms für einen bestimmten Kontext (z. B. mit Hilfe von UML-Profilen) auszuschließen (Dalgarno u. Fowler, 2008, S. 4f.). Bei einem Versuch mehr als eine Generalisierungsbeziehung zu erstellen, ausgehend von einer Klasse hin zu einer anderen, würde das Verletzen dieser Regel zu einem Hinweis für den Benutzer führen. Ebenfalls wäre es denkbar mit *default*-Werten zu arbeiten, so dass bei einer fehlenden Benutzereingabe automatisch vorher definierte Standardwerte ergänzt werden.

Regeln sind generell wie eine Whitelist zu betrachten. Dies bedeutet, dass nur das in einem Modell erlaubt ist, was spezifiziert ist, mit dem Ziel, die Handhabbarkeit von Regeln, in Bezug auf ein inkrementelles Erstellen und anschließendes Testen, zu verbessern (Kelly u. Tolvanen, 2008, S. 252). Anders verhält sich dies beispielsweise in der UML: „things are already allowed and then we start adding extra rules to override those already defined in the standard metamodel“ (Kelly u. Tolvanen, 2008, S. 252).

### 5.3.2 Modelle

Obwohl Modelle die Manifestation dessen sind, was in der Definition der DSML festgelegt wurde und damit implizit zu einer DSML gehören, so sprechen Kelly u. Tolvanen diesem Bestandteil von DSM aufgrund deren Charakteristiken eine wichtige und explizit erwähnenswerte Rolle zu. Denn Modelle im Kontext von DSM unterscheiden sich von anderen Modellen vor allem darin, dass Modellierende direkt mit den Konzepten der Domäne arbeiten und

dabei Systeme strukturell oder verhaltensorientiert modellieren (Kelly u. Tolvanen, 2008, S. 77). Außerdem zeichnen sich Modelle im Vergleich zu anderen Modellen in diesem Kontext durch weitere typische Eigenschaften aus, wie die Formalität, den höheren Grad an Abstraktion im Vergleich zur Programmierenebene und die Orientierung an der Domäne, vor allen Dingen an deren Regeln (Kelly u. Tolvanen, 2008, S. 77). Darüber hinaus sehen Kelly u. Tolvanen im Entwicklungsprozess einer DSM-basierten Lösung in Modellen viele Vorzüge. Sie sind neben einem Mittel der Kommunikation eine Chance, früh mit dem Testen von Anwendungen anzufangen und damit spät erkannte Fehler zu vermeiden, denn die Entwicklung einer DSM-basierten Lösung erfordert ein hohes Maß an iterativ und inkrementellem Vorgehen (Kelly u. Tolvanen, 2008, S. 78). Die Möglichkeit des frühen Testens gilt gleichermaßen sowohl für die Definition der Sprache, als auch für die damit verbundene Entwicklung des nachgelagerten Generators, welcher mit der DSML entwickelte Modelle in Code umsetzt. Außerdem sehen Kelly u. Tolvanen einen Vorteil darin, dass das Debuggen hauptsächlich auf Modellebene von statten geht, was bedeutet, dass Modelle in DSM dem Quellcode in herkömmlichen Softwareprojekten entsprechen und in gewisser Weise ersetzen (Stahl u. Völter, 2006, S. 16). Zudem erlaubt der Einsatz von Modellen, welche auf einer DSML basieren, dass mehrere, zielgruppenspezifische Notationen für die Modellierung angeboten werden können, welche beispielsweise auch für unterschiedliche Teile eines zu generierenden Systems verwendet werden und damit verschiedene Bausteine repräsentieren können (Tolvanen u. Kelly, 2010, S. 3). Dabei ist nach Kleppe zu erwähnen, dass textuelle Modelle sich von grafischen Modellen unterscheiden und zwar in der Art und Weise, wie die jeweiligen Tools die Modellierung unterstützen: Tools, mit denen textuelle Modelle erstellt werden können, fokussieren auf die Notation einer DSML, um Sprachentwickler dabei zu unterstützen Scanning- und Parsing-Regeln festzulegen (Kleppe, 2009, S. 42). Bei grafischen Modellen, so Kleppe, liegt der Fokus der Tools hingegen auf der abstrakten Syntax, die den Sprachentwickler unterstützen, Editoren zu entwickeln, welche gemäß der abstrakten Syntax nur die Erstellung von syntaktisch validen Modellen erlauben (Kleppe, 2009, S. 42).

### 5.3.3 *Generatoren*

Im DSM-Ansatz werden Generatoren dafür verwendet, auf einer DSML basierende Modelle in Quellcode zu transformieren, der anschließend entweder interpretiert oder kompiliert wird, ohne ein manuelles Editieren durch Entwickler zu erfordern (Kelly u. Tolvanen, 2008, S. 79f.). Im Gegensatz zu ei-

nem MDSD-Ansatz wird im DSM neben den statischen Strukturen, wie bei der UML<sup>15</sup>, auch Verhalten modelliert und anschließend generiert. Dabei hat ein Generator die Aufgaben, auf ein oder mehrere Modelle zuzugreifen, (ggf. unter Einbezug von Metadaten<sup>16</sup>) diese zu durchlaufen, dabei Daten aus Modellen zu extrahieren und die gewonnenen Daten so zu kombinieren, dass daraus Code für das Domain Framework generiert werden kann (Kelly u. Tolvanen, 2008, S. 80f.). Dieser Prozess, bei dem ein oder mehrere Modelle als Input für einen Generator dienen und Letzterer daraus Text (meist Quellcode oder aber auch Konfigurationsdateien, Dokumentation etc.) erzeugt, wird als *model-to-text* (M2T) Transformation bezeichnet und stellt die am häufigsten eingesetzte Transformationsart im Kontext von DSM dar (Combemale et al., 2017, S. 176). Dem gegenüber gestellt stehen *model-to-model* (M2M) Transformationen, welche sich im Input für einen Generator nicht von M2T-Transformationen unterscheiden, jedoch der Output des Generators ein oder mehrere Modelle sind (Combemale et al., 2017, S. 176). Sinn und Zweck letzterer Art von Transformation kann beispielsweise die Vereinfachung eines Modells in ein weiteres Modell sein, welches beispielsweise anschließend durch eine M2T-Transformation in Quellcode übersetzt wird. Ein Beispiel für diese Art der Transformation stellt die Überführung eines abstrakten Klassenmodells in ein relationales Datenbankmodell dar, bei dem das Quellmodell mit für die Datenbank relevanten Informationen, wie Primär- und Fremdschlüssel, angereichert wird und damit ein neues, weniger abstraktes Zielmodell entsteht (Pietrek u. Trompeter, 2007, S. 70f.). Eine weitere Unterscheidung bezieht sich nicht nur auf den Input und Output von Generatoren, sondern im Falle einer M2T-Transformation auf die Art und Weise, wie der Generator definiert ist. Combemale et al. und Kleppe unterscheiden in Transformationen, die auf einer Programmiersprache basieren, Template-basierte Sprachen einsetzen oder eine Kombination aus beiden verwenden (Combemale et al., 2017, S. 180ff.; Kleppe, 2009, S. 151ff.).

Bei auf Programmiersprachen basierten Generatoren muss das Traversieren des Modellinputs in einer festgelegten Reihenfolge erfolgen, da Variationen in der Reihenfolge zu verschiedenen Outputs führen können (Combemale et al., 2017, S. 180). Um eine definierte Reihenfolge einhalten zu können,

15 Die UML kann auch das Verhalten von Systemen oder deren Bestandteilen in Diagrammen erfassen, jedoch ist hier die Menge an Diagrammen im Vergleich zu einem DSM basierten Ansatz wesentlich höher, um Systemverhalten ausreichend zu spezifizieren (Kelly u. Tolvanen, 2008, S. 82).

16 Beispielsweise können zu Metadaten gehören: Die Bildschirmposition von Modellierungskonzepten in einem Modell, Zeitpunkt der Erstellung, Modellversion oder Annotationen in einem Modell (Kelly u. Tolvanen, 2008, S. 81).



wird oftmals auf das Verhaltensmuster *Visitor (Besucher)* (Gamma et al., 1995, S. 366ff.) zurückgegriffen. Es erlaubt eine Unterteilung in Elemente, welche in einem Modell vorkommen, und deren Besucher, die Operationen auf den Modellelementen ausführen können und zwar ohne dabei bestimmte Operationen nur für bestimmte Klassenhierarchien zuzulassen. Jedoch gibt es ebenso eine Variante, bei der das Visitor Design Pattern nicht manuell umgesetzt werden muss, da ein Framework, wie Eclipse Modeling Framework (EMF) existiert, welches ein Application Programming Interface (API) für den Zugriff auf ein Inputmodell bereitstellt. Der Quellcodeauszug 1 zeigt hierzu ein Beispiel<sup>17</sup>. In der ersten Phase wird das Modell `model.smvcm1`<sup>18</sup> unter Zuhilfenahme der EMF API geladen, in der zweiten Phase ein Objektgraph aufgebaut und die Modellelemente sequentiell traversiert und dabei Attribute und Methoden in Zeichenketten erzeugt, um in der dritten Phase die in der Variable `code` befindlichen Daten mit Hilfe eines File-Streams in eine Java Datei zu schreiben.

Code 1: Beispiel für die Generierung von Java-Quellcode mit Hilfe eines in Java geschriebenen Programms

```

1 // PHASE 1
2 ResourceSet resourceSet = new ResourceSetImpl();
3 Resource resource = resourceSet.getResource(URI.create("model.smvcm1"));
4
5 // PHASE 2
6 TreeIterator allElementsIter = resource.getAllContents();
7 while (allElementsIter.hasNext()){
8     Object object = allElementsIter.next();
9     if (!object instanceof Class) continue;
10    Class cl = (Class) object;
11
12    // String variable for collecting code statements
13    String code = "package entities;\n\n",
14    code += "import java.io.Serializable;\n\n";
15    code += "public class " + cl.getName() + " implements Serializable{\n"
16        ;
17
18    // generate Attributes:
19    Iterator <Attribute> attIter = cl.getAtts (); //... code += ...
20
21    // generate Methods:
22    Iterator <Operation> opIter = cl.getOps (); //... code +=
23    //... code += "};";

```

17 Entnommen aus (Brambilla et al., 2012, S. 129f.).

18 EMF wird in XML bzw. XML Metadata Interchange (XMI) serialisiert, weshalb die Modelldaten bereits vorstrukturiert sind. XMI stellt ein Referenzaustauschformat für Modelle dar, dessen Einsatz von einem Konsortium von Werkzeugherstellern forciert wird. XMI enthält keine grafischen Angaben (z. B. Position oder Größe von Modellelementen), sondern lediglich inhaltlich-strukturelle Informationen.

```

24 // PHASE 3
25 try {
26     FileOutputStream fos = new FileOutputStream(cl.getName() + ".java");
27     fos.write(code.getBytes());
28     fos.close();
29 }
30 catch (Exception e) { }
31 }

```

Bei Template-basierten Sprachen hingegen werden vordefinierte Sprachkonstrukte (ebenfalls eine domänenspezifische Sprache) mit Modellinformationen angereichert, welche anschließend von einer Template Engine an die entsprechenden Stellen eingesetzt werden (Combemale et al., 2017, S. 184). Combemale et al. verstehen unter dem Begriff Template ein „[...] raw piece of an artifact with explicitly marked expressions that are evaluated relative to a model and whose results are inserted to complete the artifact“ (Combemale et al., 2017, S. 184). Bei dieser Art der Transformation sind folglich drei verschiedene Sprachen vorzufinden: Die Zielsprache, also die Sprache des Generators, die Templatesprache, welche die Sprachkonstrukte der Templates und deren Verwendung festlegt sowie die Abfragesprache, die schließlich für die Extraktion und Aufbereitung der Daten aus dem Modell genutzt wird (Combemale et al., 2017, S. 184). Der Quellcode 2 zeigt beispielhaft, wie die drei Sprachen in *Eclipse Acceleo*<sup>19</sup> (Obeo GmbH, 2017a), einem quelloffenen, auf dem Standard Meta Object Facility Model to Text (MOFM2T) basierten Template-Generator umgesetzt werden. Die erste Zeile, die Deklaration eines Templates, erinnert an eine Deklaration einer Methode in einer objektorientierten Programmiersprache wie Java. Die darauffolgende Kontrollstruktur [ `if ( . . . )` ] ist, wie die Template-Deklaration, Teil der Templatesprache von Acceleo, während jedoch der boolsche Ausdruck der `if`-Anweisung zur Abfragesprache Acceleo Query Language (AQL) gezählt wird, eine auf OCL basierende Sprache von Acceleo. In der darauffolgenden Zeile drei ist die Zielsprache Java in der Deklaration eines Hinweises zu erkennen, mit anschließender Instanziierung. Dazwischen befinden sich AQL-Ausdrücke, die eingebettet in die Template-syntax sind und bei Ausführung der Template Engine durch entsprechende Informationen aus dem Modell ersetzt werden.

Code 2: Auszug aus einer Acceleo-Defintion zur Generierung von Java-Code

```

1 [template public erstelleHinweis (aufgabe : Aufgabe)]
2 [if(aufgabe.loesungstipp.hinweistext.toString().equalsIgnoreCase('
   invalid')._not())]

```

<sup>19</sup> Das Konzept von Template-basierten Transformationen wird anhand von Acceleo erläutert, da dieser Generator auch später für die Umsetzung verwendet wurde.

```

3   Hinweis [aufgabe.loesungstipp.sichtbarerBezeichner/] = new Hinweis("[
    aufgabe.loesungstipp.hinweistext/"]);
4   [aufgabe.loesungstipp.sichtbarerBezeichner/].setHinweisId([aufgabe.
    loesungstipp.id/]);
5   [aufgabe.sichtbarerBezeichner/].setLoesungstipp([aufgabe.loesungstipp.
    sichtbarerBezeichner/]);
6 [/if]
7 [/template]

```

Der Quellcodeauszug 3 zeigt den Output des Generators in der Zielsprache Java aus Quellcode 2. Die im Template definierten Chars und Zeichenketten werden direkt in den Output übernommen, während alle AQL-Anweisungen interpretiert und geparsed werden müssen und schließlich automatisiert erstellten Java Quellcode ergeben, welcher beispielsweise in einer Methode eingebettet sein kann.

Code 3: Java Code Output des Beispieltemplates

```

1 Hinweis Hinweis_19571 = new Hinweis("Hoeren Sie sich den Podcast ab
    Minute 1:40 nochmal etwas genauer an!");
2 Hinweis_19571.setHinweisId(1396618125);
3 SingleChoice_381.setLoesungstipp(Hinweis_19571);

```

#### 5.3.4 Domain Framework

Als vierten und letzten Bestandteil des DSM-Ansatzes sehen Kelly u. Tolvanen das Domain Framework. Sie verstehen darunter die Schnittstelle zwischen generiertem Code und der darunter liegenden Zielumgebung (Kelly u. Tolvanen, 2008, S. 86, 311; Völter et al., 2013, S. 94). Dabei kann der generierte Code direkt als Input bzw. Trigger für das Funktionieren eines Systems sein oder eine Schnittstelle zur Zielumgebung nutzen bzw. vom Code des Domain Frameworks genutzt werden (nach dem Vorbild des Hollywood-Principles<sup>20</sup>) (Kelly u. Tolvanen, 2008, S. 86, 320f., 325f.). Durch die Bereitstellung von Code im Domain Framework wird Funktionalität, welche sonst in der DSML bzw. dem Generator implementiert werden müsste, in das Framework verlegt. Dadurch kann der Generator funktional schlanker und weniger komplex gehalten werden. So kann ein Domain Framework beispielsweise „[...] utility functions, data structures to be instantiated, ways to attach behavioral code generated from the models [...]“ (Kelly u. Tolvanen, 2008, S. 318) bereitstellen.

<sup>20</sup> Das Hollywood-Principle (Sweet, 1985, S. 216) besagt „Don’t call us, we will call you“ und spielt damit auf das Architekturprinzip Inversion of Control (IoC) an, bei dem die Steuerung über die Ausführung von Code einer Anwendung auf das Framework, welches sie umgibt und bei dem sie bekannt ist, verlagert wird (Geirhos, 2015, S. 584f.).

Auf welche Art und Weise ein Domain Framework erzeugt wird und ob dies überhaupt benötigt wird, hängt nach Strembeck u. Zdun von den folgenden Faktoren ab: die Anforderungen an den Generator und die Zielumgebung, die Ausdrucksfähigkeit der DSML sowie die Verfügbarkeit von existierenden bzw. noch benötigten Frameworks (Strembeck u. Zdun, 2009, S. 1264).

Die Herausforderung an die Erstellung einer DSM-basierten Lösung besteht darin, dass jedes Konzept einer Domäne einer der drei Schichten DSML, Generator oder Domain Framework zugeordnet werden muss. Völter et al. geben dazu die Empfehlung, dass Konzepte, die unterschiedlich in einer Domäne ausgestaltet sein können, in die DSML integriert werden, welche Abstraktionen dieser Konzepte bereithält (Völter et al., 2013, S. 29). Unveränderliche Konzepte, welche für jede Anwendung in einer Domäne identisch sind, werden im Domain Framework verarbeitet (Völter et al., 2013, S. 29). Und jene Domänenkonzepte, welche von starren Regeln der Anwendung abgeleitet werden können, die mit der DSML ausgedrückt und dabei nicht gleich in jeder Anwendung angewendet werden, aber deren Struktur in der DSML gleich ist, sollen in einem Generator implementiert werden (Völter et al., 2013, S. 29).

#### 5.4 VORÜBERLEGUNGEN ZUR ERSTELLUNG DER DOMAIN-SPECIFIC MODELING LANGUAGE

Obwohl für diese komplexen Fragestellungen der Erfassung und Zuordnung von Domänenkonzepten zu den Bestandteilen von DSM, und im Speziellen für die Definition einer DSML, nur wenige formale Qualitätskriterien und allgemeine Designkriterien existieren (Frank, 2013, S. 134), soll aus Gründen der Nachvollziehbarkeit zunächst definiert werden, wie bei der Erstellung der DSML vorgegangen wurde.

##### 5.4.1 Vorgehen

Den Empfehlungen von Combemale et al., Kelly u. Tolvanen und Den Haan folgend wurden die folgenden Schritte durchlaufen<sup>21</sup> (Combemale et al., 2017, S. 34; Kelly u. Tolvanen, 2008, S. 341ff.; Den Haan, 2009):

<sup>21</sup> Das Vorgehen könnte den Eindruck eines Wasserfallmodells erwecken, was jedoch auf die Entwicklung von DSM-basierten Lösungen nicht zutrifft. Vielmehr ist ein hohes Maß an iterativ-inkrementeller Entwicklung vonnöten, welches bei der Erstellung auch angewandt wurde.

1. *Definition der Ziele und Anwendungsbereiche der DSML*: Initial soll festgelegt werden, welchem Zweck eine zu erstellende DSML dient, wie der Anwendungsbereich und damit die spätere Zieldomäne eingegrenzt wird und welche Qualitätskriterien an die DSML es zu erfüllen gilt (Den Haan, 2009).
2. *Analyse der Quelldomäne auf Basis von Nutzungsbeispielen*: In diesem Schritt, soll die Problem-domäne systematisch analysiert werden, um daraus konzeptuelle Anforderungen an die DSML ableiten zu können. Combemale et al. empfehlen hierzu, sich an (abstrakten) Beschreibungen oder Beispielen aus der Problem-domäne stammend, zu orientieren (Combemale et al., 2017, S. 34).
3. *Definition der abstrakten Syntax*: Aus dem vorangegangenen Schritt sollen Domänenkonzepte und deren Eigenschaften abgeleitet werden, welche die Kernklassen eines Metamodells und damit die abstrakte Syntax der DSML bilden (Kelly u. Tolvanen, 2008, S. 342). Zugehörig soll die Semantik der einzelnen Konzepte festgelegt werden (Den Haan, 2009).
4. *Beziehungen zwischen den Konzepten des Metamodells definieren*: Die abstrakte Syntax soll nun mit Beziehungen zwischen den Konzepten angereichert werden, um der DSML eine Struktur zu geben (Combemale et al., 2017, S. 35). Wie auch bei der Definition der abstrakten Syntax sind Architekturprinzipien, wie eine hohe Kohäsion (Starke, 2014, S. 169) der Konzepte und lose Kopplung (Starke, 2014, S. 168) durch die Beziehungen zueinander zu beachten (Kelly u. Tolvanen, 2008, S. 342).
5. *Metamodell einschränken bzw. erweitern*: Das Metamodell soll anhand von (kontextabhängigen) Constraints einschränkt werden, so dass jede Modellinstanz daraus das Metamodell korrekt repräsentiert (Combemale et al., 2017, S. 35). Darüber hinaus soll das Metamodell durch Informationen erweitert werden, welche die Variation von Konzepten in Modellen zulassen (Kelly u. Tolvanen, 2008, S. 342). Zudem soll das Metamodell durch Informationen erweitert werden, die beispielsweise sicherstellen, dass eine Umsetzung einer Modellinstanz durch einen Generator korrekt funktioniert (Kelly u. Tolvanen, 2008, S. 342).
6. *Erstellung der Notation*: Basierend auf der abstrakten Syntax soll eine Notation erstellt werden, welche für Benutzer einen hohen Wiedererkennungswert aufweist und einem Domänenkonzept eindeutig zugeordnet werden kann, bzw. nur genau ein Domänenkonzept grafisch repräsentiert (Kelly u. Tolvanen, 2008, S. 342f.).

7. *Werkzeugunterstützung erstellen*: Es sollen Tools<sup>22</sup> erstellt werden, die die Erstellung von Modellen mit der DSML ermöglichen und dabei (Korrektheits-) Regeln aus der Domäne berücksichtigen (vorher bereits definiert durch Constraints im Metamodell), um sicherzustellen, dass Modellierende ausschließlich valide Modelle erstellen (Kelly u. Tolvanen, 2008, S. 342).

Die Frage der Zuordnung der Domänenkonzepte setzt voraus, dass diese bekannt sind. Deshalb wurde das folgende Vorgehen bei der Definition der DSML angewandt und dabei den Empfehlungen von Combemale et al. (2017, S. 34) Den Haan (2009), Kleppe (2009, S. 43) und Frank (2013, S. 136, 140ff.) gefolgt. Der Einbezug von verschiedenen Design Empfehlungen für die Erstellung von DSM-basierte Lösungen rührt daher, dass ein etabliertes Prozessmodell quasi nicht existent ist und das Vorgehen dabei auf unterschiedliches Verständnis in einschlägigen Arbeiten stößt (Frank, 2013, S. 133f.).

#### 5.4.2 *Zweck und Anwendungsbereich*

Aus der Literatur wurden in Kapitel 4 bereits Forschungsdesiderate aufgezeigt, für deren Bearbeitung noch keine ausreichend ausgereifte Technologie vorhanden ist (Dichev u. Dicheva, 2017; Dicheva et al., 2015; Dicheva u. Dichev, 2016). Diese Lücke soll durch die Entwicklung einer DSML und weiterer Bestandteile einer DSM-basierten Lösung geschlossen werden. Mit der DSML und der entsprechend Unterstützung von Werkzeugen soll es möglich sein, spielifizierte Lehr-Lernarrangements zu modellieren und diese für Lernende im Rahmen einer Anwendung und unter Berücksichtigung eines didaktischen Konzepts *erlebbar* zu machen. Dies würde nicht-technisch versierten Benutzern erlauben, ein Technologie-unterstütztes Konzept wie Gamification in Lehr-Lernkontexten effizient einzusetzen, ohne dafür weitreichendes technisches Verständnis zu benötigen oder gar Entwicklungstätigkeiten durchführen zu müssen. Dies wird im Bereich der (Serious-) Game Authoring Tools schon länger als gewinnbringend empfunden (Mehm et al., 2012; Mehm et al., 2013). Konkret könnte mit einer derartigen DSM-Lösung beispielsweise untersucht werden, wie sich der Kontext auf den Erfolg eines Einsatzes eines spielifizierten Lehr-Lernarrangements auswirkt, indem mit der DSML erstellte Modelle zwischen Lehrenden ausgetauscht und in verschiedenen Kontexten als Arrangement erprobt werden (Bartel et al., 2017a, S. 1501). Durch die Anleitung

<sup>22</sup> Häufig auch unter *Language Workbench* (Combemale et al., 2017), als Werkzeug zur Definition der DSML und *Solution Workbench* (Den Haan, 2009; Fowler, 2010), als Werkzeug zur Anwendung der DSML in der Literatur zu finden.

im Rahmen des GCDP bzw. den Vorüberlegungen, die im GCDR *manuell* durch Lehrende getroffen werden, ist ein Maß an Standardisierung erreicht, welches zumindest aus methodischer Sicht eine Vergleichbarkeit gewährleisten würde. Die während der Nutzung der spielifizierten Anwendung durch die Lernenden anfallenden Daten (z. B. in Bezug auf Nutzungsdauer, Fortschritt bei der Bearbeitung etc.) können im Sinne von Learning Analytics sowohl Rückschlüsse auf die Effektivität des spielifizierten Lehr-Lernarrangements zulassen als auch die Lernmotivation von Lernenden verbessern. Die Verbesserung des Lernens soll ein zentrales Ziel der DSM-Lösung sein. Im Rahmen dieser Arbeit wurde eine vollständige DSM-Lösung namens *Emendo* (lat.: verbessern) erarbeitet.

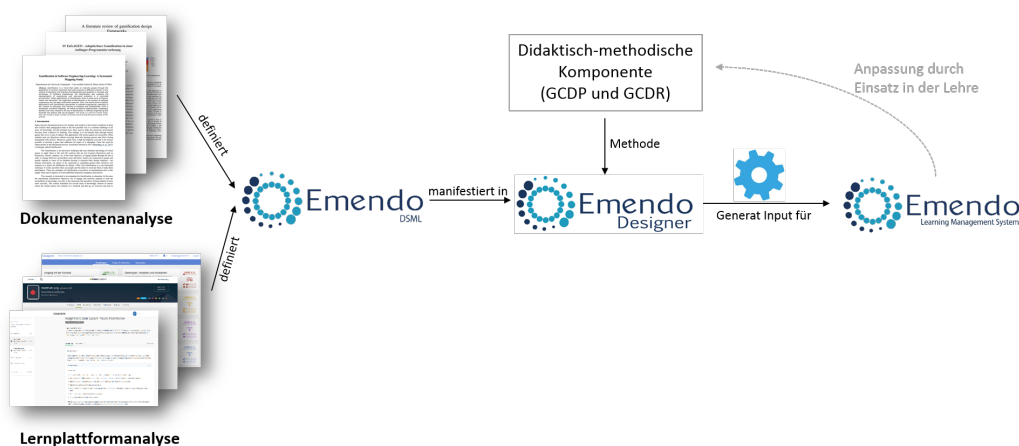


Abbildung 34: Überblick über die Bestandteile von Emendo DSM

Die Abbildung 34 zeigt die Bestandteile von Emendo im Überblick. Die Emendo DSML wird auf Basis der beiden durchgeführten Analysen erstellt und im Emendo Designer umgesetzt. Mit Letzterem wird unter Einbezug des GCDP und des GCDR ein Modell eines spielifizierten Lehr-Lernarrangements erstellt, welches automatisch durch einen Generator in eine lauffähige Instanz eines Emendo LMS (Domain Framework) umgesetzt werden soll. Das Emendo LMS wird von Lernenden im Rahmen von Lehrveranstaltungen verwendet und soll durch die darin integrierte Funktionalität Rückschlüsse auf die didaktisch-methodische Komponente zulassen, um die modellierten spielifizierten Lehr-Lernarrangements zu verbessern.

#### 5.4.3 Anforderungen an die Emendo DSML

Frank schlägt vor, sich generischer Anforderungen (beispielsweise aus einem generischen Anforderungskatalog stammend (Frank, 2013, S. 142)) zu bedie-

nen, um den pragmatischen Charakter und die Qualität einer DSML im Vorfeld durch Kriterien abzusichern (Frank, 2013, S. 136f.). Dieser Vorschlag geht mit der Forderung von DSR einher, wonach das Vorgehen im Rahmen einer iterativen Artefaktentwicklung eine Festlegung von (Qualitäts-) Anforderungen an das Artefakt vorsieht, die schließlich durch evaluative Verfahren validiert werden sollen. Frank nennt generische Anforderungen an DSMLs, die auch für die Emendo DSML gelten sollen und in daran angepasster Formulierung wie folgt lauten (Frank, 2013, S. 136f.):

- **Semantische und grafische Affordance** (Norman, 1995): Die Domänenkonzepte in der Emendo DSML sollen die Quelldomäne semantisch eindeutig wiedergeben, indem in der Quelldomäne verwendete Terminologie in der DSML rekonstruiert wird. Eine entsprechende Notation, soll das Verständnis der Konzepte für die Nutzer der DSML erhöhen. Daraus folgt, dass Modelle, die mit der DSML erstellt werden, intuitiv und leicht zu verstehen sein sollen sowie im Umkehrschluss die Emendo DSML eine leichte Erlernbarkeit für Domänenexperten bieten soll.
- **Semantische Eindeutigkeit:** Die Emendo DSML soll bezogen auf deren Domänenkonzepte und auf den Anwendungsfokus semantische Eindeutigkeit aufweisen, d.h. jedes Konzept besitzt im Anwendungsfokus der DSML genau nur eine semantische Bedeutung. Dies soll ebenfalls zu einem intuitiv-logischen Umgang beim Modellieren für Domänenexperten beitragen.
- **Erweiterbarkeit:** Die Emendo DSML soll mit Änderungen in der Sprachdefinition umgehen können und entsprechende Erweiterungsmöglichkeiten für die Integration neuer Domänenkonzepte bereitstellen.
- **Separierte Abstraktionslevel:** Beim Modellieren mit der Emendo DSML sollen klar voneinander getrennte und für Benutzer wahrnehmbare Abstraktionsgrade der Sprache verwendet werden. Demnach soll für Benutzer ersichtlich sein, welche Abstraktionsstufen durch welche Domänenkonzepte repräsentiert werden.
- **Eindeutigkeit von Sprachkonzept zu Zielkonzept:** Die Emendo DSML soll alle benötigten Informationen in den Modellen enthalten, die für eine Umsetzung in die jeweiligen Zielkonzepte benötigt werden. Auf den eigentlichen Umsetzungsprozess sollen Benutzer keinen Einfluss haben, sodass semantische Variationen zwischen Sprachkonzept und Zielkonzept ausgeschlossen sind.



Bezogen auf die konkreten Anforderungen wurde durch die Analyse der Problemdomäne mit einer durchgeführten Dokumenten- und Lernplattformanalyse eine konzeptuelle Basis geschaffen, die es in die zu erstellende Sprache zu übertragen gilt. Dabei sind insbesondere die Konzepte aus Tabelle 15 gemeint, die unter Berücksichtigung der oben genannten generischen Anforderungen als Basis für die Erstellung der abstrakten Syntax (inkl. den Beziehungen der Konzepte) dienen sollen. Die abstrakte Syntax soll im Folgenden fokussiert werden.

## 5.5 ABSTRAKTE SYNTAX

Das Metamodell, welches die abstrakte Syntax der Emendo DSML darstellt, basiert inhaltlich auf den Konzepten aus Tabelle 15 und deren vorangegangenen Beschreibungen. Es zeigt außerdem das Ergebnis des 3. und 4. Schritts, es beschreibt also neben der abstrakten Syntax aus Gründen der Nachvollziehbarkeit ebenso die Beziehungen zwischen den Elementen. Die Abbildung 35 zeigt das Metamodell der Emendo DSML in Form eines UML-Klassendiagramms<sup>23</sup>. Das grundlegende Paradigma, welches bei der Konzeption des Metamodells umgesetzt ist, ist die Tatsache, dass die Struktur der Elemente der DSML von der Verwendung der Elemente innerhalb des Emendo LMS unabhängig ist. Die Dynamik des Emendo LMS wird zwar in der Emendo DSML strukturell beschrieben, jedoch obliegt die Ausführung dem LMS. Dieses Paradigma gilt für alle Elemente der Emendo DSML. Zunächst sollen die strukturellen Elemente der Sprache beschrieben werden, um anschließend die Konzepte zur Erstellung von Dynamik einführen zu können.

Zu beachten ist, dass bei dem Emendo DSML Metamodell nicht alle Beziehungen zwischen den Elementen dargestellt sind, da diese sich teilweise auf den Ebenen der Kindklassen befinden, welche in weiteren Modellen detailliert werden. Insgesamt umfasst das Metamodell mehr als 160 abstrakte und konkrete Klassen. Sie sollen im Folgenden unter Bezugnahme auf die konzeptuellen Anforderungen auszugsweise beschrieben werden.

---

<sup>23</sup> In den UML-Klassendiagrammen, welche die Emendo DSML beschreiben und detaillieren, wurde aus Gründen der Übersicht auf die Angabe von Standardzugriffsmethoden für Attribute und Konstruktoren verzichtet. Gemäß dem objektorientierten Prinzip der Datenkapselung (auch bekannt als Information Hiding - siehe z. B. Starke (2014, S. 127f.) oder Meyer (2007, S. 25.)) besitzen alle modellierten Attribute die Sichtbarkeit *private* und ein Zugriff darauf erfolgt über Standardzugriffsmethoden. Wird im Text Bezug auf Diagrammbestandteile, wie Klassennamen oder Attributnamen genommen, so werden diese kursiv gekennzeichnet und sprachlich dem Lesefluss angepasst, also ggf. auch konjugiert.

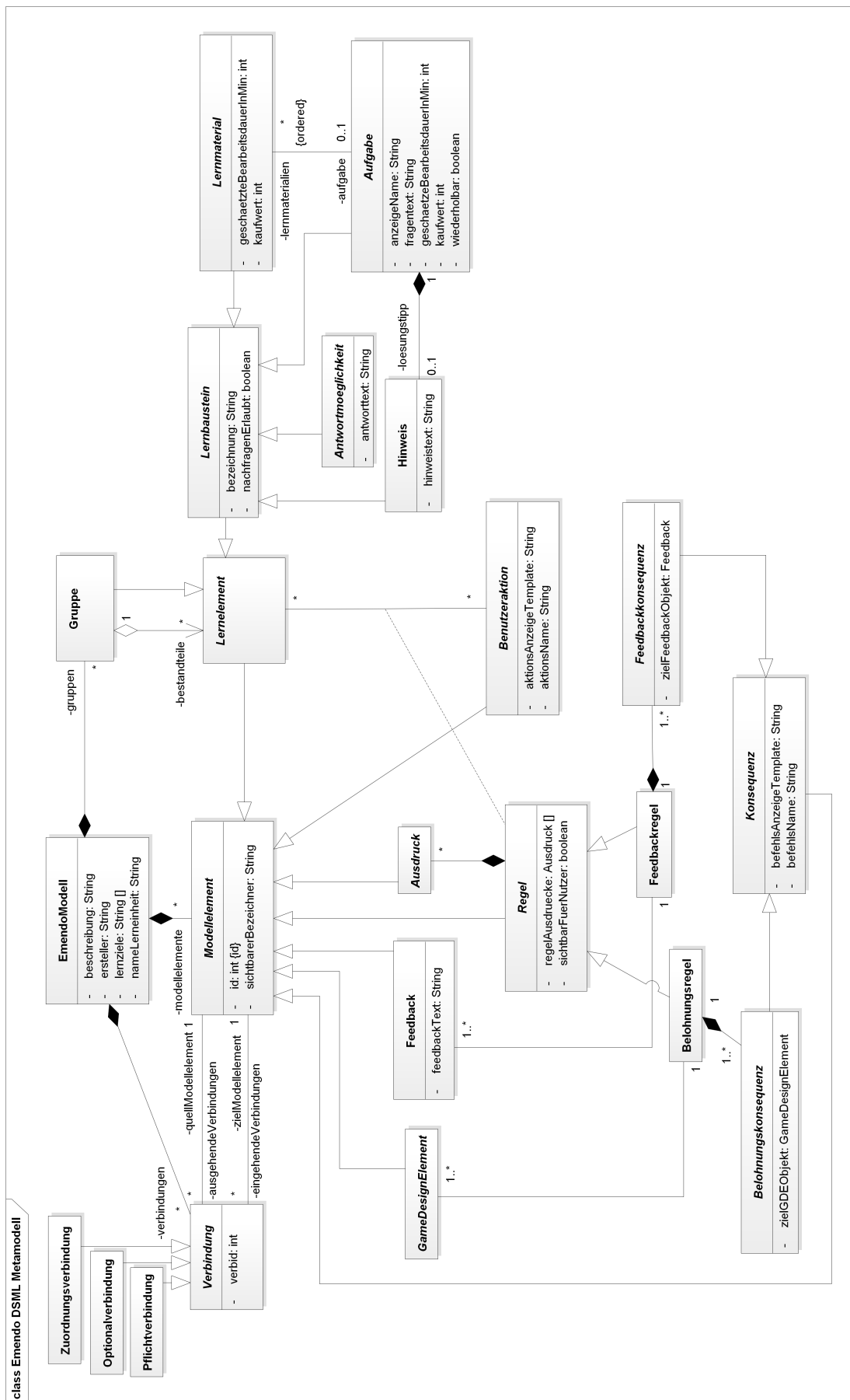


Abbildung 35: Das Metamodell der Emendo DSML

### 5.5.1 *EmendoModell*

Jedes Modell, welches eine Instanz der DSML darstellt, wird durch die zentrale Klasse *EmendoModell* repräsentiert. Diese Container-Klasse vereint alle zum Modell gehörenden Elemente und stellt darüber hinaus Attribute bereit, welche die Modellinstanz und damit die spielifizierte Lerneinheit weiter detaillieren. So besitzt jede Instanz eine Beschreibung der spielifizierten Lerneinheit, den Namen der erstellenden Person, damit verknüpfte Lernziele (LE\_A1), wie im GCDP definiert und im GCDR ausformuliert sowie einen aussagekräftigen Namen (LE\_A2). Ein *EmendoModell* besteht zudem aus Modellelementen, welche durch eine innerhalb einer Modellinstanz eindeutige id identifiziert und mit dem Attribut *sichtbarenBezeichner*, als Teil der Notation, für die modellierende Person benannt werden können.

### 5.5.2 *Verbindung und Gruppe*

Modellelemente werden mit dreierlei Verbindungen miteinander verknüpft, wobei jedes Modellelement mehrere ausgehende und eingehende Verbindungen besitzen kann und jede Verbindung genau ein Quell- und Zielmodellelement hat (siehe Abbildung 36). Pflicht- und Optionalverbindungen lassen sich zwischen Lernbausteinen anwenden (A1 und B1), wohingegen Zuordnungsverbindungen Modellelemente anderen Modellelementen zuordnen (z. B. Lernbaustein A1 zu Belohnungsregel C1), da sie in einem bestimmten Abhängigkeitsverhältnis zueinander stehen oder Informationen unidirektional (im Falle einer Zuordnung von einem Hinweis zu einer Aufgabe) oder bidirektional (Aufgabe zu Regel) voneinander benötigen. Eine Pflichtverbindung sagt aus, dass um B1 bearbeiten zu können, A1 bearbeitet sein muss. Eine Optionalverbindung lässt diese Abhängigkeit offen und setzt die Bearbeitung von A1 nicht voraus, um B1 zu bearbeiten. Es kann A1 oder B1 bearbeitet werden. Welche Zuordnungsverbindungen zwischen A1 und C1 bestehen dürfen, bedarf einer Einschränkung auf Typebene. Diese wurde mit OCL im *Emendo Designer* umgesetzt und ist damit nicht mehr Teil der abstrakten Syntax, wird jedoch als Teil des *Emendo Designers* beschrieben.

Gruppen hingegen, gruppieren Lernelemente im Sinne eines Kompositums (Gamma et al., 1995, S. 183ff.). Sie bestehen rekursiv entweder aus weiteren Gruppen oder Lernelementen. Dabei ist jedes Lernelement immer genau einer Gruppe zugeordnet. Dies bedeutet, dass die Gruppe\_o aus Abbildung 36 die Wurzelgruppe bildet, in der weitere Lernelemente enthalten sein können. Eine Belohnungsregel als Beispiel für C1 darf hingegen nur außerhalb einer

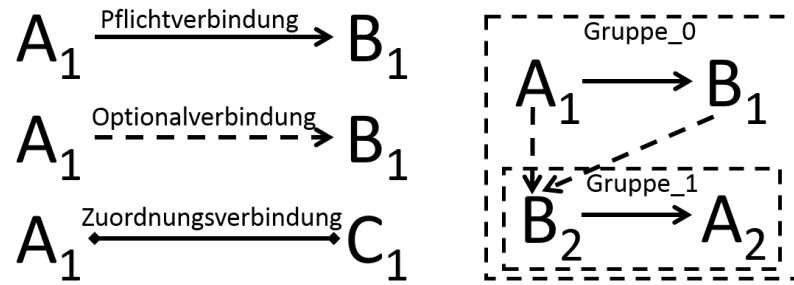


Abbildung 36: Verbindungen in Emendo DSML

Gruppe liegen, kann jedoch einer Gruppe zugeordnet werden, da es bei einer Gruppe um ein Lernelement handelt.

Gruppen leisten als rein strukturelle Elemente einen wesentlichen Beitrag zur Erzeugung von Spielifizierung, vor allem durch die Möglichkeit Mechanics im Sinne von Werbach u. Hunter (2012) zu definieren. Sie sind für Lehrende zur Strukturierung der spielifizierten Lehr-Lernarrangements unabdingbar, jedoch für Lernende transparent. So ist es beispielsweise mit der Hilfe von Gruppen möglich, Lernbausteine derart zu strukturieren, dass sie einen *Quest* mit entsprechendem Überblick und Instruktion (siehe *Q\_A1* bzw. *Q\_A2*) als Teil einer Erzählung (siehe *Q\_Z1*) bilden, der beispielsweise ein bestimmtes Lernziel realisieren soll. Ein *Quest* kann dabei nur aus einer einzelnen Aufgabe bestehen oder eine Sammlung von Lernbausteinen beinhalten, die entweder einzeln oder gesamt belohnt werden (siehe *Q\_R1*) können. Weiterhin ist es denkbar, dass Gruppen dazu genutzt werden, Mechanics, wie Challenges oder Competition umsetzen, da beispielsweise zum Abschluss einer Gruppe eine herausfordernde Aufgabe als Challenge gelöst oder möglichst viele Badges pro Gruppe gesammelt werden sollen, um eine Competition mit Mitlernenden zu gewinnen.

### 5.5.3 Lernelement und Lernbaustein

Modellelemente können diverse Spezialisierungen besitzen. Eine davon sind *Lernelemente*, welche – wie bereits zuvor definiert – zum Lernen innerhalb von Lernaktivitäten verwendet werden. Sie lassen sich weiter spezialisieren in *Gruppen*, die rein strukturell-logische Bestandteile der Emendo DSML repräsentieren sowie *Lernbausteine*. Letztere besitzen eine Bezeichnung (vererbt in Kindklassen gemäß *A\_A1*, *V\_A1*, *PC\_A1*, *T\_A1*) und ein Flag *nachfragenErlaubt*, das definiert, ob von Lernenden Fragen zu einem bearbeitbaren Lern-

baustein gestellt werden können. Dieses Attribut ist die Konsequenz daraus, dass in den Subtypen der abstrakten Klassen *Aufgabe* (A\_A9) und *Lernmaterial* (V\_A3, PC\_A3 und T\_A3) die Möglichkeit bestehen sollte, im Kontext von Lernaktivitäten Fragen stellen zu können. Als Standardwert dieses Attributs wird in jeder Instanz des Metamodells *true* gesetzt. Für den Fall, dass *Aufgaben* im Kontext von Leistungsaufgaben eingesetzt werden sollen, um beispielsweise den Lernfortschritt von Lernenden zu prüfen, muss es möglich sein, die Nachfragen bei derartigen Aufgaben zu unterbinden. In einem solchen Fall, wird der Attributwert schlichtweg in einer entsprechenden Instanz negiert.

#### 5.5.4 *Lernmaterial*

Die Klasse *Lernmaterial* ist eine abstrakte Kindklasse von Lernbaustein und besitzt ein Attribut *geschaetzteBearbeitungszeitInMin*, in welchem Lehrende die vollständige Bearbeitungszeit für ein konkretes Lernmaterial angeben können. Die Existenz dieses Attributs, setzt die Anforderung LE\_A3 indirekt um, da die Gesamtbearbeitungszeit für eine Lerneinheit für Lernende ersichtlich sein soll. Demnach wird diese nicht als Teil der DSML, sondern als abgeleitetes Attribut im Domain Framework aus der Summe der einzelnen konkreten *Lernmaterialien* und konkreten *Aufgaben* berechnet. Des weiteren hält das Lernmaterial ein Attribut *kaufwert*. Dieses standardmäßig mit 0 initialisierte Attribut, ermöglicht es die Anforderung CU\_Z1 zu realisieren, wonach *Lernmaterialien* (oder auch *Aufgaben*) mit einem Wert >0 charakterisiert werden können, was dem Kaufwert in Points (im Speziellen *Redeemable Points* (Zichermann u. Linder, 2013, S. 19)) entspricht, um dieses Element durch einen Lernenden freizuschalten. Eine Freischaltung ist im Domain Framework so definiert, dass diese nur möglich ist, wenn der Punktestand von Lernenden größer oder gleich dem Kaufwert eines *Lernmaterials* oder einer *Aufgabe* ist. Diese Art der Freischaltung von *Lernmaterialien* oder *Aufgaben* ist eine Alternative zu einem Einsatz von *Verbindungen*.

Die Abbildung 37 stellt die drei konkreten Kindklassen von *Lernmaterial* dar. Es handelt sich um *Video* und *Podcast*, welche beide gleichermaßen ein Attribut besitzen, womit eine *URL* angegeben werden kann, die auf das Quellmedium zeigt. Zwar ist in beiden Fällen ein Player, wie in Anforderungen V\_A2 und PC\_A2 vorgesehen, nicht direkter Bestandteil der DSML, jedoch werden diese Anforderungen indirekt durch diese Attribute zur Spezifizierung der *URL* umgesetzt. Die dritte konkrete Kindklasse ist *Textblock*, welche die Definition von Zeichenketten im Attribut *inhalt* (T\_A2) zulässt und derartige Blöcke

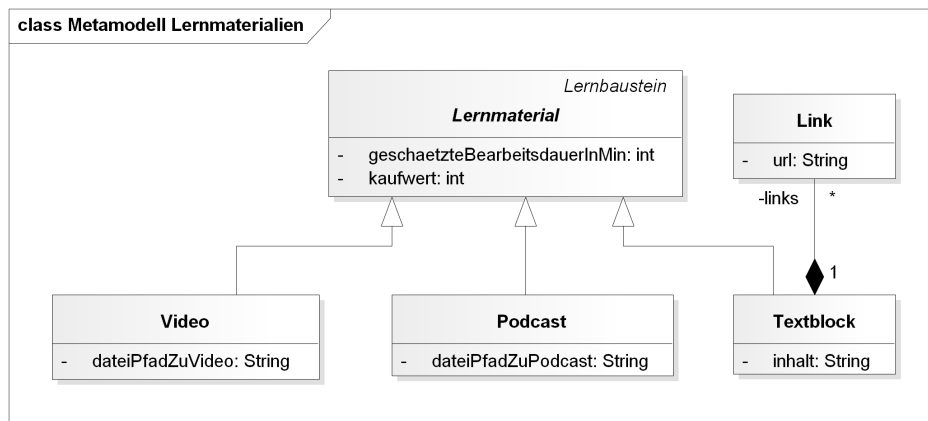


Abbildung 37: Metamodell inkl. konkreter Ausprägungen zu Lernmaterialien

URLs in Form von *Links* beinhalten können. Dies ist zum Beispiel bei der Angabe von Internetquellen der Fall, oder wenn Medien, wie Abbildungen, direkt in Textteilen referenziert werden sollen.

Konkrete *Lernmaterialien* können entweder in *Aufgaben* eingebettet werden (V\_Z1, PC\_Z1, T\_Z1) oder als alleiniger *Lernbaustein* bestehen. In ersterem Fall mag die Reihenfolge der Einbettung in *Aufgaben* entscheidend sein, da Lernende beispielsweise innerhalb einer Aufgabenbearbeitung zunächst textuellen und anschließend visuellen Input in Form eines *Videos* benötigen, bevor sie eine *Aufgabe* absolvieren. Diese Möglichkeit und die damit einhergehende geforderte Flexibilität soll durch eine entsprechende geordnete Referenz von *Aufgabe* zu *Lernmaterial* strukturell gesichert sein.

### 5.5.5 Aufgabe

Eine weitere Spezialisierung von *Lernbaustein* ist die abstrakte Klasse *Aufgabe*. Sie besitzt neben vererbten Attributen ein Attribut *fragentext* (A\_A2), in welcher die Aufgabenstellung spezifiziert werden kann. Das Attribut *anzeigenName* wird dafür verwendet, ein Mapping von interner Repräsentation in Daten auf für Anwender lesbare Bezeichnungen zuzulassen. Diese lesbaren Bezeichnungen sind als Standardwerte gesetzt. Für eine *Aufgabe* kann im Attribut *wiederholbar* angegeben werden, ob diese wiederholbar sein soll oder nicht. Eine Wiederholung einer *Aufgabe* würde implizieren, dass diese zuvor bereits ein Mal vollständig bearbeitet worden ist. Bei einer Wiederholung würden alle vorherigen Ergebnisse, sowie mit der Bearbeitung verknüpften *Regeln* rückgängig gemacht. Die Ergebnisse aus Dokumenten- und Lernplattformanalyse lassen sich in Bezug auf *Aufgaben* anhand deren Antwortformate

(A\_A3) unterscheiden. Dabei kristallisieren sich vier verschiedene Aufgabentypen heraus, die in Abbildung 38 dargestellt sind.

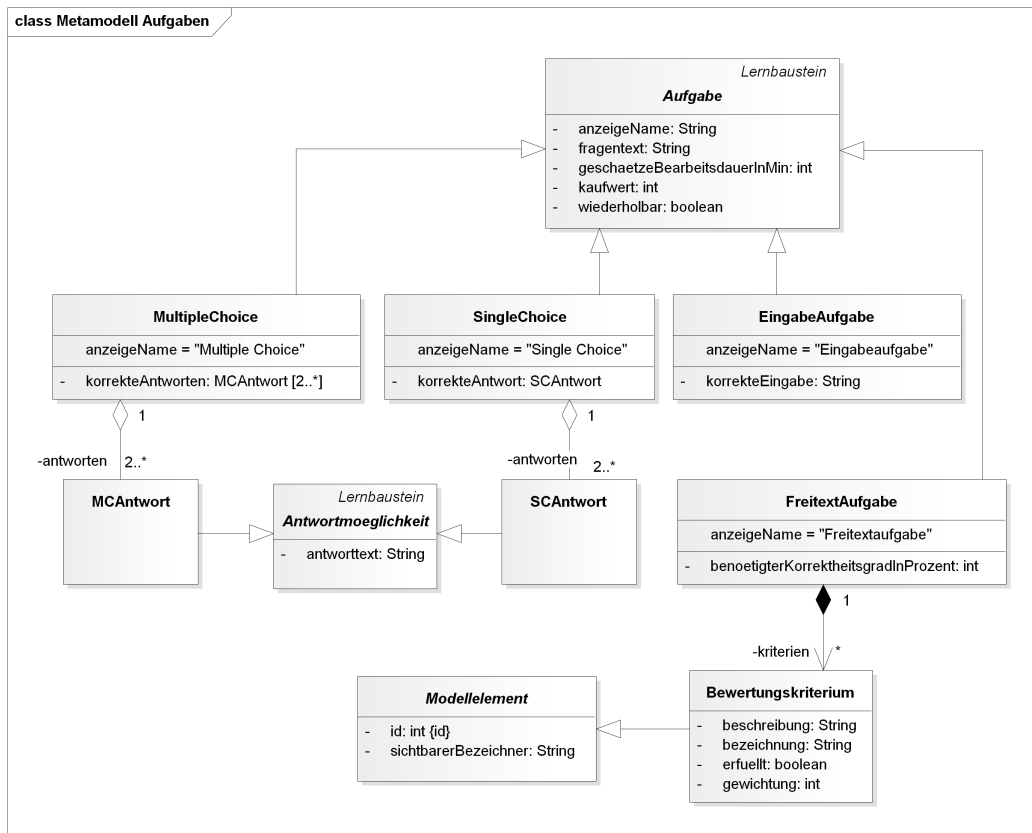


Abbildung 38: Metamodell zu Ausprägungen von Aufgaben

*Multiple Choice*-Aufgaben besitzen mindestens zwei Antwortmöglichkeiten, konkret vom Typ *MCAntwort*, von denen die korrekten Antworten in der Liste *korrekteAntworten* gespeichert werden.

Ähnlich verhält es sich bei *Single Choice*-Aufgaben, bei denen es ebenfalls mindestens zwei *Antwortmöglichkeiten* gibt, in diesem Fall vom Typ *SCAntwort* und von denen nur genau eine *SCAntwort* als korrekte Antwort im Attribut *korrekteAntwort* gesetzt werden kann.

Eine *EingabeAufgabe* stellt einen weiteren Typ dar. Es handelt sich um einen Aufgabentyp, bei dem die Antwort in Form eines einzelnen Begriff oder einer Zahl getätigt werden kann (siehe z. B. Warm-Up Aufgaben bei JiTT). Eine entsprechende Möglichkeit die korrekte Lösung zu einer *Aufgabe* zu hinterlegen, bietet sich durch das Attribut *korrekteEingabe* an.

Der vierte und letzte Aufgabentyp ist die *FreitextAufgabe*. Sie wird durch die Eingabe eines literarischen Textes oder von Quellcode gelöst. Derartige Eingaben werden durch *Bewertungskriterien* beurteilt. Beispielsweise soll es möglich sein einen Text anhand der *Bewertungskriterien* fachliche Korrekt-

heit, Originalität der Lösung, Umfang und Ausdrucksweise beurteilen zu können. Hierfür wird jedes der einzelnen *Bewertungskriterien* eigens beschrieben (Attribut: *beschreibung*) und erhält eine Bezeichnung (Attribut: *bezeichnung*). Zudem kann angegeben werden, ob ein Kriterium eine vom Faktor 1 abweichende Gewichtung erhalten soll (Attribut: *gewichtung*), welche bei der Verrechnung der Kriterien (nach dem arithmetischen Mittel) mit dem Attribut *benoetigterKorrektheitsgradInProzent* verglichen wird und schließlich zu einer richtigen oder falschen Aufgabenlösung führt. Zu jedem Bewertungskriterium wird erfasst, ob es erfüllt wurde oder nicht. Standardmäßig steht der Wert dieses Attributs *erfuellt* auf false.

Zu allen Aufgaben kann zudem ein Hinweis (A\_A8) angegeben werden, der Lernende bei der Aufgabenlösung unterstützt.

Da es sich bei einem *Quest* auch nur um einzelne Aufgabe handeln kann, sind ebenso die Anforderungen Q\_A4 (freiwillige Bearbeitung von *Quests* ist abhängig von dem Einsatz von Pflicht- und Optionalverbindungen) und Q\_A5 (mehrfache Bearbeitung eines *Quests*, da alle Aufgaben können wiederholbar sein) erfüllt.

#### 5.5.6 Regel

Bei den im Folgenden beschriebenen *Regeln*, handelt es sich aus Sicht von DSM um Domänenregeln. Korrektheits- und Konsistenzregeln werden im Zuge des Emendo Designers beschrieben, dem Werkzeug, welches die Emendo DSML umsetzt und für Benutzer das Erstellen von Modellen erlaubt, die auf der Emendo DSML basieren und letztere instanziiieren. Regeln in der Emendo DSML beschreiben das dynamische Verhalten von *Lernelementen* sowie die damit verknüpften spielifizierten Elemente strukturell. Jede *Regel* wird hierfür genau einem *Lernelement* zugeordnet, wobei sich *Regeln* dabei entweder auf *Lernbausteine*, wie *Aufgaben* und *Lernmaterialien* oder auf *Gruppen* beziehen (festgelegt durch OCL im Emendo Designer). Zudem enthält jede *Regel* eine *Benutzeraktion*, mindestens einen *Ausdruck* (in der Rolle als Bedingung), dessen Überprüfung – wenn positiv – mindestens eine *Konsequenz* zur Ausführung bringt.

##### 5.5.6.1 Benutzeraktion

Auf Subtypen von *Lernelementen* können *Benutzeraktionen* ausgeführt werden. Diese Aktionen transformieren die Zustände von *Lernelementen* und führen unter bestimmten Bedingungen zu bestimmten *Konsequenzen*. In der Emen-



do DSML besteht jede *Benutzeraktion* aus einem *aktionsNamen*, welcher der Klassenname ist und für die Erzeugung der Klasse mittels Reflection (Ullendörff, 2017, S. 898ff.) im Domain Framework verwendet wird, sowie einem *aktionsAnzeigeTemplate* (siehe Abbildung 39). Der Grund für letzteres Attribut liegt in der Art und Weise wie Benutzer die Definition von *Regeln* vornehmen sollen, bei denen *Benutzeraktionen* ein fester Bestandteil sind. So soll, abhängig ob sich eine *Benutzeraktion* auf einen *Lernbaustein* oder auf eine *Gruppe* bezieht, bei der Definition einer neuen *Regel* das konkrete Bezugsobjekt (Instanz von *Gruppe* oder *Lernbaustein*) in einem Template dynamisch angezeigt werden.

Beispielhaft wird aus Systemsicht eine *Benutzeraktion Aufgabenbearbeitung-Begonnen* als Teil einer Regeldefinition instanziiert, bei der die Vorbelegung des Attributs *aktionsAnzeigeTemplate* den String Die Bearbeitung von <Aufgabe> wird begonnen enthält. In <Aufgabe> soll bei der Definition einer *Benutzeraktion*, die für eine konkrete Instanz einer *Aufgabe* definiert wird, der *sichtbareBezeichner* der *Aufgabe* stehen, also zum Beispiel <SCA\_38145>. Dies soll zu einem intuitiven Umgang mit der Sprache für Benutzer beitragen. Wie in Abbildung 39 zu erkennen, existieren eine Reihe an abstrakten Subklassen von *Benutzeraktion* bzw. deren Kindklasse *LernbausteinBenutzeraktion*. Diese beziehen sich auf Aufgaben, welche entweder *Freitextaufgaben* oder keine *Freitextaufgaben* darstellen, also manuell oder automatisch bewertet werden sowie den in der DSML definierten *Lernmaterialien Video*, *Textblock* (hier repräsentiert durch *Links*, da auf *Textblöcken* als Ganzes keine sinnvollen Aktionen durch Benutzer ausgeführt werden können) und *Podcasts*. Das Clustering von konkreten *Benutzeraktionen* anhand von abstrakten Elternklassen ist nicht nur der Tatsache geschuldet, dass nicht alle *Benutzeraktionen* gleichermaßen auf *Lernbausteinen* ausgeführt werden können, sondern vielmehr ist, wie an anderen Stellen bei der Sprachdefinition, auf die Erweiterbarkeit der Emendo DSML Rücksicht genommen worden, indem neue *Benutzeraktionen* schlichtweg durch die Erstellung einer neuen und zum Elterntyp passenden Spezialisierung eingefügt werden können. Des weiteren wurde darauf geachtet, dass alle Klassen-Cluster eine hohe Kohäsion (Starke, 2014, S. 169) besitzen sowie dem Offen-Geschlossen-Prinzip (Starke, 2014, S. 169f.) folgen, welches besagt, dass Komponenten geschlossen für Änderungen aber offen für Erweiterungen sein sollen (Starke, 2014, S. 169). Erwähnenswert ist zudem auch, dass wenn Attribute existieren, die ein Zielobjekt referenzieren, wie beispielsweise das Attribut *ziellinkObjekt* der abstrakten Klasse *LinkBenutzeraktion* dies damit zu begründen ist, dass die Auswahl, auf die sich eine Benutzeraktion (oder in den folgenden Erläuterungen auch eine

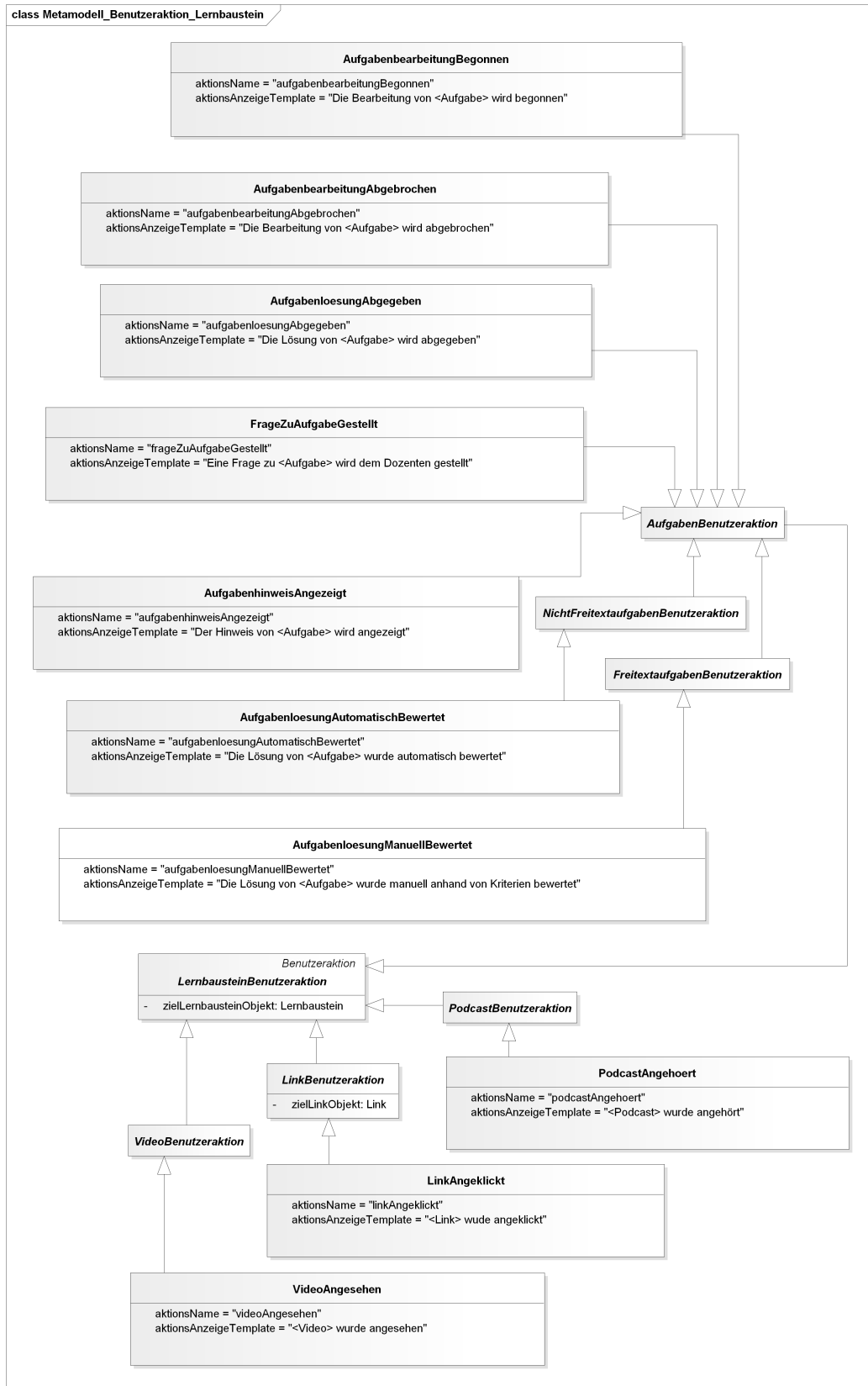


Abbildung 39: Auf Lernbausteine bezogene Benutzeraktionen in Emendo DSML

Variable oder eine Konsequenz) bezieht nicht eindeutig ist, da mehrere Instanzen als Bezugsobjekte möglich sind. Das Beispiel des *zielLinkObjekt*-Attributs veranschaulicht diesen Sachverhalt, da ein *Textblock* mehrere *Links* beinhalten kann, jedoch nur ein konkreter *Link* die *LinkAngeklickt Benutzeraktion* triggern soll. Die Existenz von *zielTypObjekt*-Variablen ist im Folgenden an mehreren Stellen in Klassendiagrammen aufgrund der selben Begründung vorzufinden.

Neben *Benutzeraktionen* die direkt auf *Lernbausteinen* ausgeführt werden können, gibt es das Pendant dazu, auf *Gruppen*. Die Ausführung erfolgt dabei nicht direkt auf dem *Lernelement Gruppe*, sondern auf *Lernelementen*, die zu einer *Gruppe* gehören, für die eine *Benutzeraktion* gelten soll. Daher ist es nicht verwunderlich, dass vergleichbare *Benutzeraktionen*, die sich auf *Lernbausteine* beziehen, ebenso im Kontext von *Gruppen* vorzufinden sind (siehe Abbildung 40). Dies ermöglicht dass *Benutzeraktionen* auch für Elemente einer *Gruppe* getriggert werden können, in dem die *Benutzeraktion* sich nicht mehr auf eine konkrete Instanz eines *Lernbausteins* bezieht, sondern vielmehr auf dessen Typ, der innerhalb einer *Gruppe* liegt. Der Unterschied wird anhand eines Beispiels deutlich: Eine *VideoAngesehen Benutzeraktion* würde getriggert werden, wenn ein *Video* von einem Lernenden vollständig angesehen wurde. Dabei würde die *Regel* direkt aus Sicht des Systems die konkrete *Lernmaterial* Instanz eines *Videos* referenzieren.

Wenn die Referenz einer *Regel* sich nicht auf die Instanz eines *Videos* bezieht, sondern auf dessen umliegende *Gruppe*, würde es sich nicht um eine *VideoAngesehen Benutzeraktion* handeln, sondern vielmehr um ein Objekt von *VideoAngesehenGruppe*. Dies hätte zur Folge, dass die *Benutzeraktion* getriggert werden würde, wenn mindestens ein *Video* innerhalb einer *Gruppe* angesehen wurde.

#### 5.5.6.2 Ausdrücke

Zur Definition einer *Regel* gehört ebenso die Festlegung eines oder mehreren *Ausdrücke*, die entscheiden, ob die *Konsequenzen* der *Regel* ausgeführt werden. Wie in Abbildung 41 dargestellt, sind Ausdrücke<sup>24</sup> im Kontext der DSML entweder *Boolsche*-, *Aufgabentyp*- oder *Ganzzahl-Ausdrücke*.

Bei boolschen Ausdrücken handelt es sich entweder um *BoolscheBedingungsvariablen*, *BoolscheLiterale* (Wahrheitswerte) oder *BoolscheBedingungen*.

<sup>24</sup> Auf weitere Operatoren, wie den unären Negationsoperator als *LogischerOperator* oder den Prozentoperator als *ArithmetischerOperator* wurde verzichtet. Eine Erweiterung um weitere Operatoren ist jedoch aufgrund der vorherrschenden Klassenhierarchie problemlos möglich.

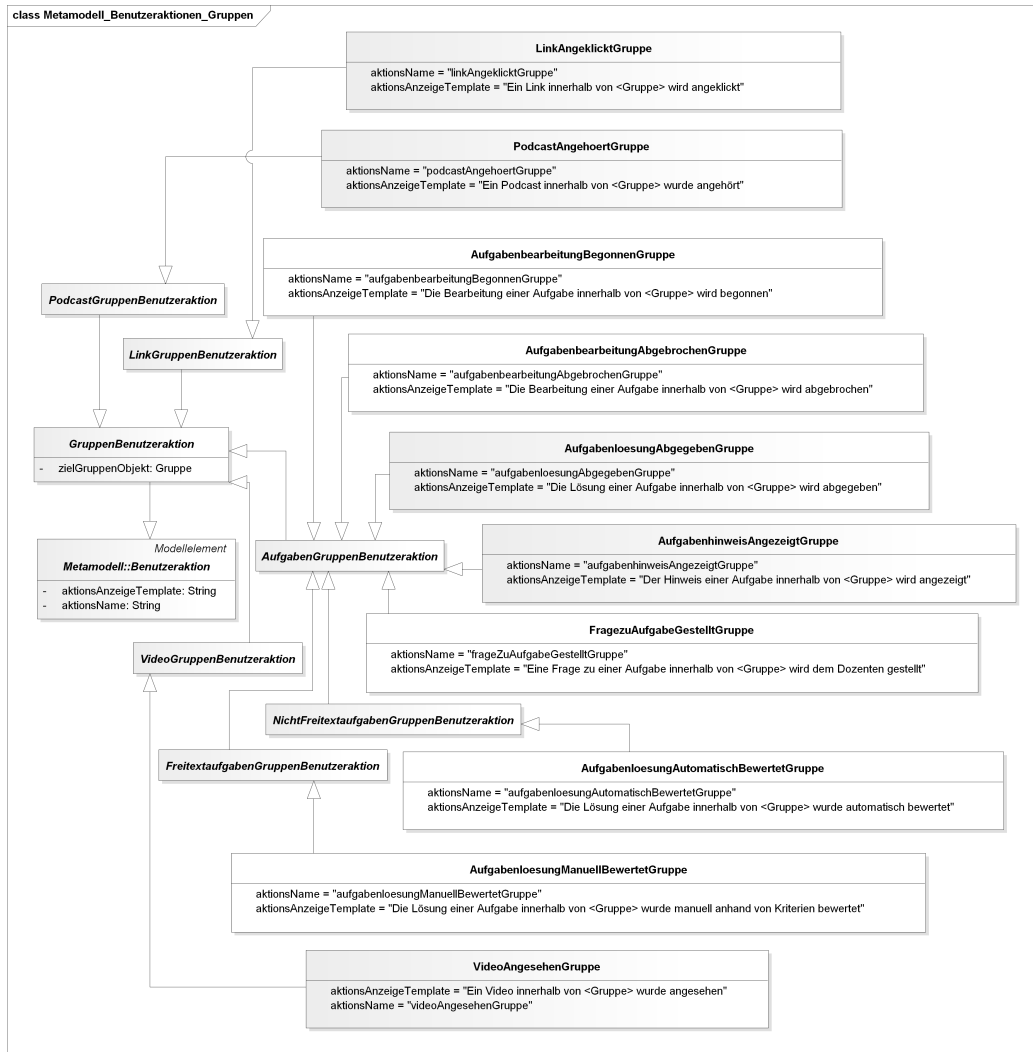


Abbildung 40: Auf Gruppen bezogene Benutzeraktionen in Emendo DSML

Als Ergebnis liefern demnach alle *BoolscheAusdrücke* bei deren Interpretation einen Wahrheitswert (*BoolschesLiteral*). *BoolscheBedingungsvariablen* nutzen, wie die bereits vorgestellten *Benutzeraktionen*, einen Template-Mechanismus für deren Anzeige und unterscheiden deshalb bei ihren Attributen in einem Variablennamen, was dem Klassennamen entspricht und dem Template, um eine Variable einem Nutzer lesbar anzuzeigen. Das Konzept von Variablen findet sich bei boolschen, ganzzahligen oder solchen Variablen wieder, welche sich auf Aufgabentypen beziehen. *Bedingungsvariablen* bietet im Allgemeinen die Möglichkeit, Variablen zu definieren, die bei der Interpretation eines Ausdrucks situativ zur Laufzeit durch das Domain Framework ausgewertet werden. So sind *BoolscheVariablen* in der Emendo DSML definiert, die es beispielsweise erlauben, zu überprüfen, ob zur Laufzeit ein bestimmtes *Unlock* bereits vom Benutzer freigeschaltet oder eine *Aufgabe* korrekt gelöst wurde

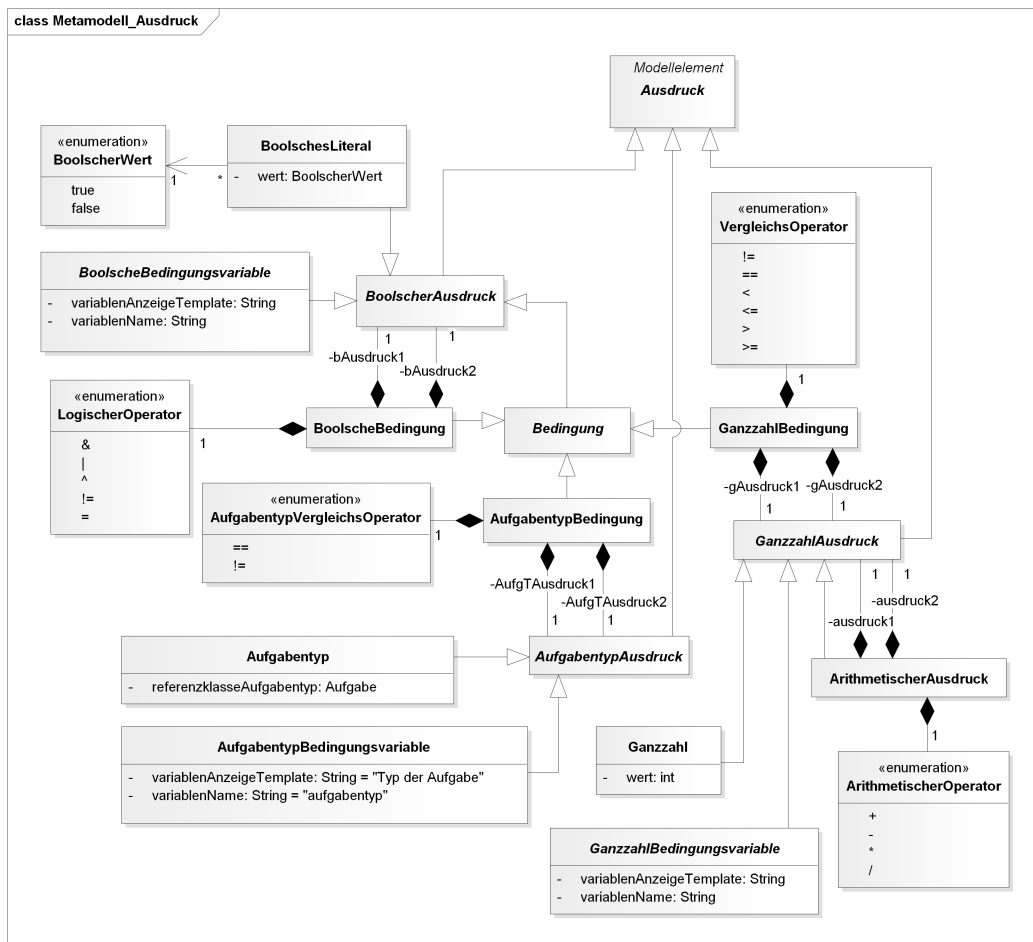
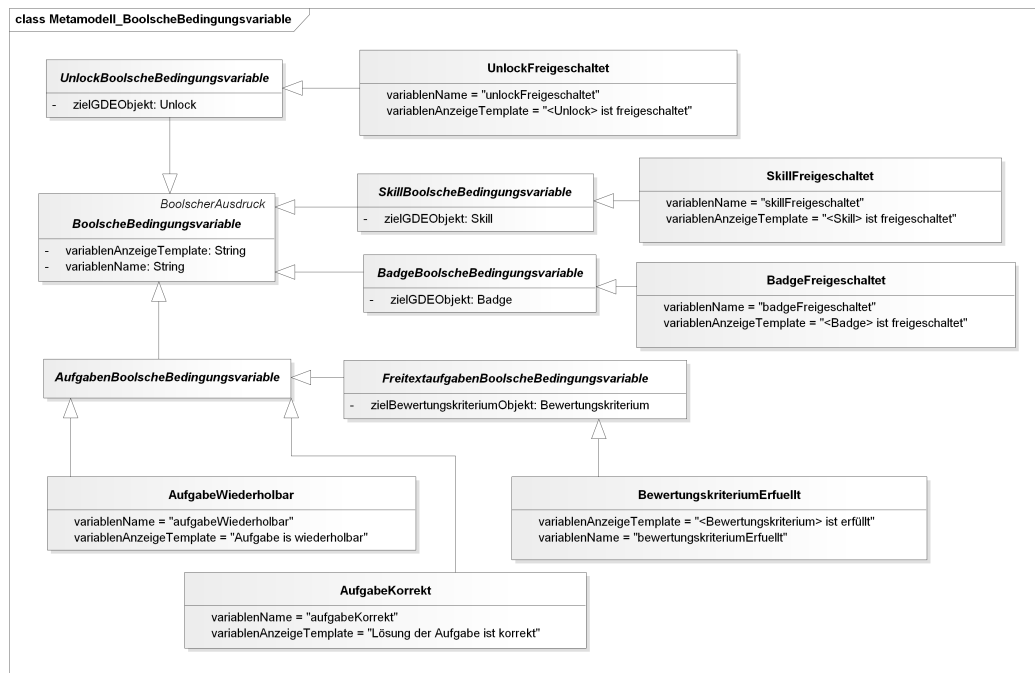


Abbildung 41: Ausdrücke von Regeln der Emendo DSML

(siehe Abbildung 42). Variablen werden immer im Kontext eines konkreten Benutzers ausgewertet. Betreffen die Überprüfungen von Variablen *GameDesignElemente*, die mehr als ein Mal durch eine *Regel* referenziert werden können (z. B. wenn an einer *Regel* als *Belohnungskonsequenz* mehr als ein *GameDesignElement* hängt, welche als *Konsequenz* einer Aktion ein Benutzer erhalten soll), muss es möglich sein anzugeben, auf welches konkrete Objekt sich eine Variablenüberprüfung bezieht.

Hierfür wurden, wie bereits erläutert und wie in Abbildung 42 zu erkennen, Referenzen auf die jeweiligen Zielobjekte eingefügt, die bei *GameDesignElementen* vom konkreten Typ des zu referenzierenden *GameDesignElements* sind (z. B. Attribut *zielGDEObjekt* vom Typ *Unlock*), sowie bei *Bewertungskriterien* eine Instanz eines Kriteriums (*zielBewertungskriterium* vom Typ *Bewertungskriterium*) referenzieren. Für *Lernbausteine*, wie *Aufgaben*, im Kontext von Variablen ist dies nicht der Fall, da in diesem Fall die Zielobjekte durch die Zuordnung einer *Regel* zu einem *Lernbaustein* bereits bekannt sind.



### Abbildung 42: Boolsche Bedingungsvariablen

*BoolscheLiterele* besitzen einen Wert, der entweder `true` oder `false` ist und damit das Ergebnis eines *BoolschenAusdrucks* repräsentieren.

Die dritte Kindklasse eines *BoolschenAusdrucks*, ist eine *Bedingung*. Alle *Bedingungen*, sind nach der Form `<Ausdruck> <Operator> <Ausdruck>` aufgebaut und besitzen, je nach Bedingungstyp, unterschiedlich zulässige Werte für die allgemeine Form und die darin enthaltenen Platzhalter. So sind beispielsweise logische Operatoren nur für *BoolscheBedingungen* zulässig, während *ArithmetischeOperatoren* nur Gültigkeit für *GanzzahlAusdrücke* besitzen. Jedoch ist allen gemein, dass *Bedingungen* sich durch *Ausdrücke* rekursiv verschachteln lassen, ohne Begrenzung der Rekursionstiefe. Dies ermöglicht eine komplexe Bedingungserstellung, die aufgrund ihrer Komplexität auch eine entsprechend hohe Ausdrucksmächtigkeit besitzt.

Neben *BoolschenBedingungen* existieren *AufgabentypBedingungen*. Sie vergleichen den Typ einer *Aufgabe* anhand eines Referenzaufgabentyps (siehe Abbildung 43). Beispielsweise können damit Bedingungen formuliert werden, die sich nur auf *SingleChoice* oder *MultipleChoice* Aufgaben innerhalb einer *Gruppe* beziehen.

Zudem unterstützt die Emendo DSML die Formulierung von *GanzzahlBedingungen*. Diese sind aus *GanzzahlAusdrücken* aufgebaut, die wiederum *ArithmetischeAusdrücke* oder *GanzzahlBedingungsvariablen* sein können und mit *ArithmetischenOperatoren* verknüpft werden. Mit *ArithmetischenAusdrücken* ist es also möglich arithmetische Operationen auszuführen, mit anderen Wor-

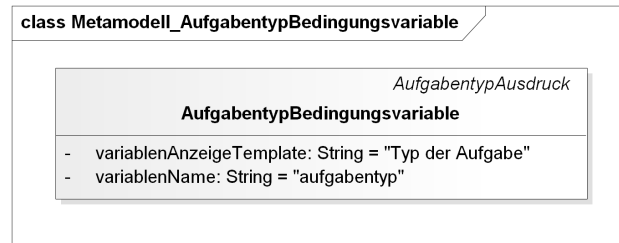


Abbildung 43: Klassendiagramm der Aufgabentyp Bedingungsvariable zum Vergleich von Aufgabentyp-Ausdrücken

ten, zur Laufzeit mit Variablen oder *GanzzahlWerten* zu rechnen. Derartige *Ausdrücke* können in *GanzzahlBedingungen* ebenfalls rekursiv eingebettet sein, ohne Begrenzung der Rekursionstiefe.

Bei *GanzzahlBedingungsvariablen* wird, wie auch schon bei *BoolschenBedingungsvariablen* in solche unterschieden, die sich auf *Gruppen* beziehen oder auf einzelne Objekte als Subtypen von *Lernelementen* oder *GameDesignElementen* (siehe Abbildung 44). So kann beispielsweise der Punktestand eines *Punktesystems* durch eine Variable abgefragt werden, ebenso wie die Anzahl aller korrekt beantworteten *SingleChoice* Aufgaben innerhalb einer *Gruppe*.

#### 5.5.6.3 Konsequenz

Ergibt die Evaluation eines jeden *Ausdrucks* einer *Regel* den *BoolschenWert true* und damit das Gesamtergebnis aller *Ausdrücke* auch *true* (*Ausdrücke* werden miteinander „verundet“), so wird mindestens eine *Konsequenz* durch das Domain Framework ausgeführt. In der Emendo DSML werden zwei verschiedene *Konsequenzen* unterschieden, die von Benutzern definiert werden können: *Feedback-* und *Belohnungskonsequenzen*.

Bei der abstrakten Klasse *Konsequenz* findet sich der bereits beschriebene Template-Mechanismus wieder. Eine *Feedbackkonsequenz*, als abstrakte Kindklasse einer *Konsequenz* hält ein *Feedback* Attribut, welches von einer konkreten Subklasse von *Feedbackkonsequenz* über Standardzugriffsmethoden überschrieben werden kann. In Abbildung 45 ist beispielsweise die *Feedbackkonsequenz FeedbackAnzeigen* zu erkennen. Wird diese Klasse als *Konsequenz* einer *Feedbackregel* instanziiert, wird das in *FeedbackAnzeigen* referenzierte *ziel-FeedbackObjekt* dem Benutzer als textuelles *Feedback* durch das Domain Framework angezeigt. Dies entspricht der Umsetzung von Anforderung A\_A4, wonach es möglich sein soll, Feedback für Lernende zu Aufgaben zu geben.

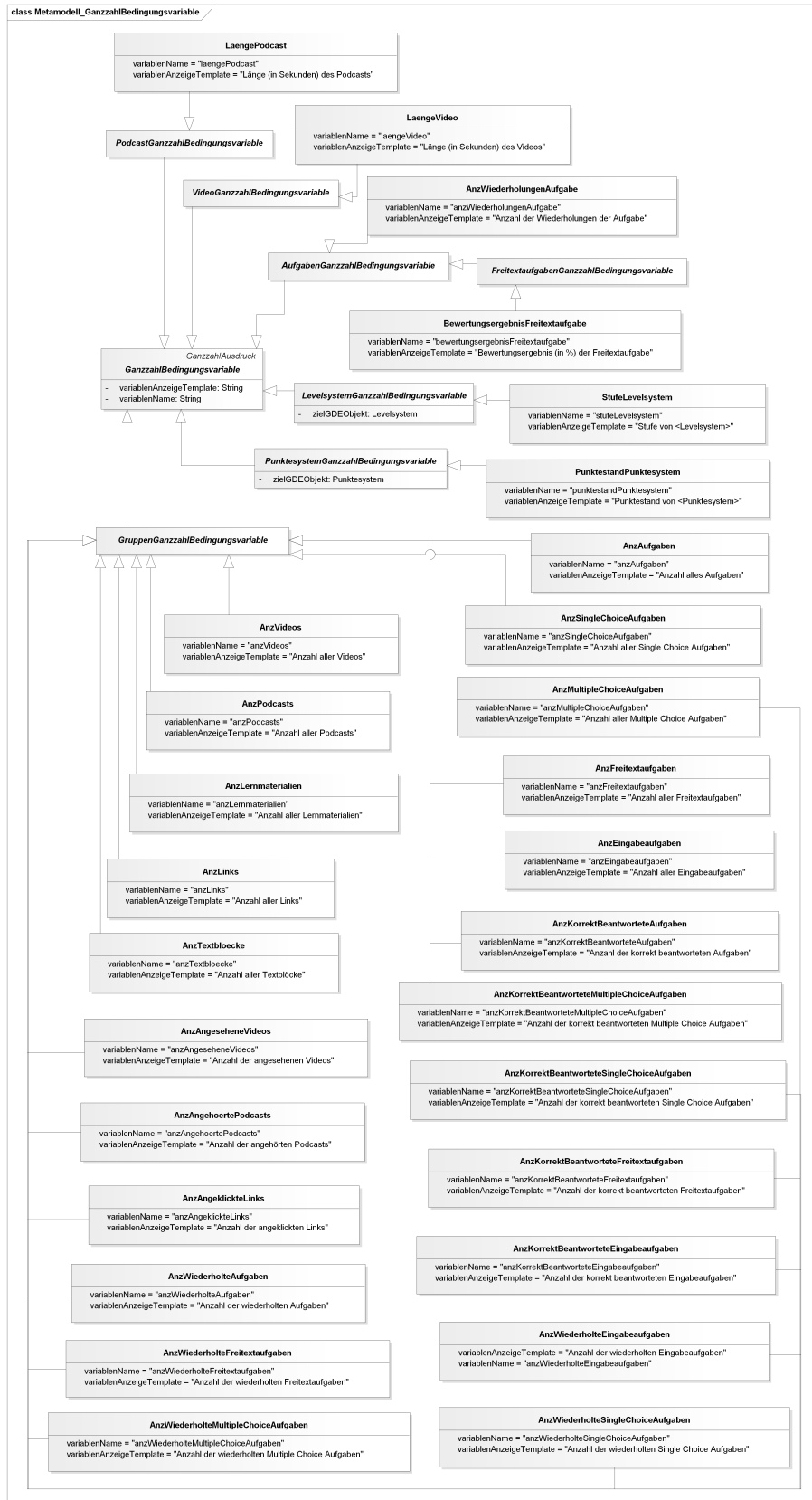


Abbildung 44: Ganzzahl Bedingungsvariablen



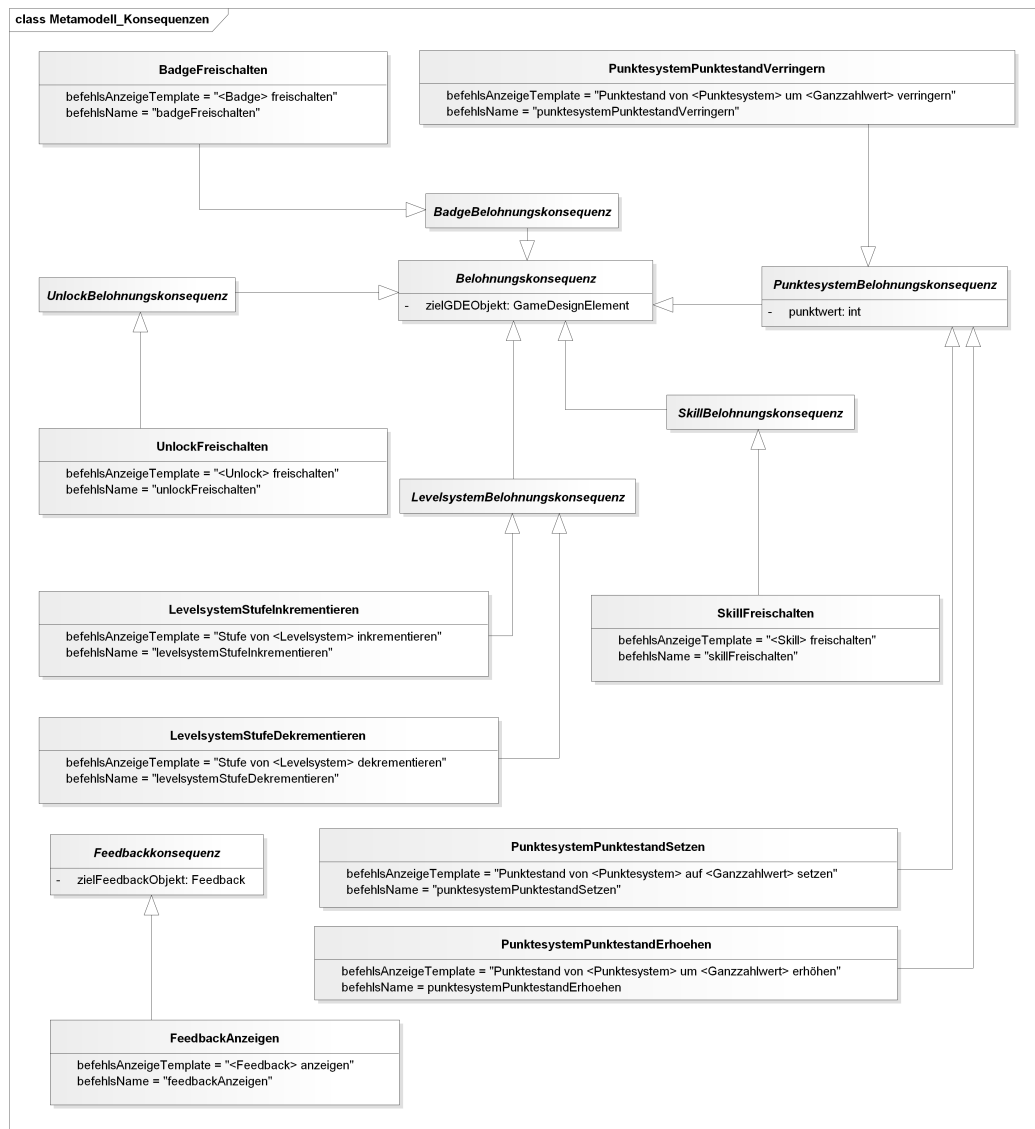


Abbildung 45: Feedback- und Belohnungskonsequenzen

Die zweite Spezialisierung einer *Konsequenz* ist die *Belohnungskonsequenz*. Diese abstrakte Klasse lässt sich weiter anhand für *GameDesignElemente* spezifische konkrete *Konsequenzen* unterteilen und besitzt, wie auch eine *Feedbackkonsequenz* ein Zielobjekt, auf das sich eine *Belohnungskonsequenz* bezieht. In dem Fall handelt es sich um *GameDesignElemente* als Typ des Attributs *zielGDEObjekt*, wie in Abbildung 46 dargestellt. Diese werden, wie in den Anforderungen B\_R1, P\_R1 gefordert, für *Benutzeraktionen* auf *Lernelementen* belohnt, die durch Lernende ausgeführt werden und bei denen definierte *Ausdrücke* zutreffen. Obwohl eine Belohnung nur direkt durch Badges und Punkte laut den Anforderungen vorgesehen ist, erscheint es sinnvoll diese Auswahl an GDEs dahingehend zu erweitern, dass Levelaufstiege oder Skills ebenso belohnt sowie neue Lernbausteine als Belohnung freigeschaltet (Con-

tentUnlocking) werden können. Im Folgenden sollen die einzelnen GDE und ihre dazugehörigen *Belohnungskonsequenzen* weiter detailliert werden:

Badges:

*Badges* sind repräsentiert durch die gleichnamige konkrete Spezialisierung von *GameDesignElement*. Sie besitzen eine *badgeBezeichnung* (siehe B\_A2), in welcher ein Kurztitel für ein *Badge* angegeben wird sowie eine *badgeBeschreibung* in der die Bedeutung des *Badges* spezifiziert wird, idealerweise unter Berücksichtigung der didaktischen Einbettung dieses GDEs. Die bildliche Repräsentation (B\_A1) wird durch das Attribut *dateipfadZuGrafik* in der Klasse *Badge* realisiert, worin eine URL angegeben werden kann, welche ein Bildformat referenziert. Ein *Badge* kann freigeschaltet werden (siehe B\_A5), was durch die Klasse *BadgeFreischalten* als Belohnungskonsequenz in der DSML umgesetzt ist.

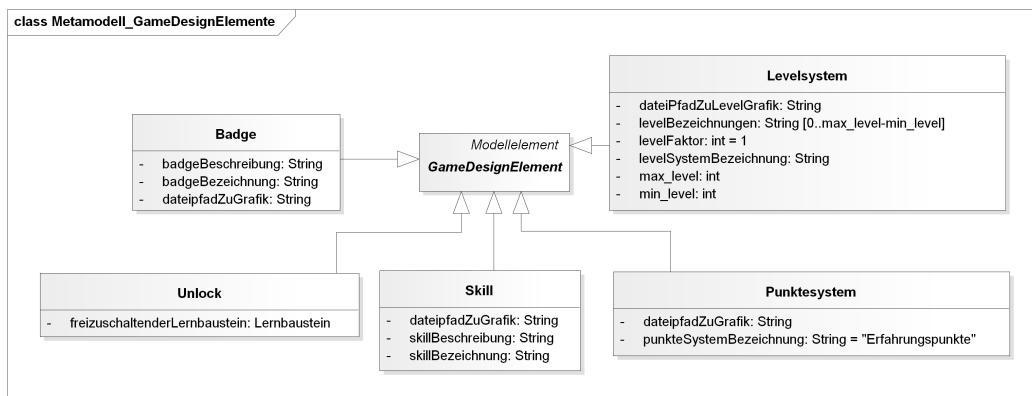


Abbildung 46: Game Design-Elemente - Components

Unlock (Content Unlock):

Die Klasse *Unlock* besitzt nur ein Attribut, in dem der freizuschaltende Lernbaustein referenziert wird. Unlocks werden durch die Klasse *UnlockFreischalten* nicht freigeschaltet, sondern ein im *Unlock* referenzierter *Lernbaustein* (siehe A\_A5, V\_A4, PC\_A4 und T\_A4). Im Kontext der Emendo DSML können demnach Subtypen von *Aufgaben* sowie von *Lernmaterialien* mit einem *Unlock* freigeschaltet werden (siehe CU\_Z2).

Skill:

Ein *Skill* ist vergleichbar zu einem *Badge* aufgebaut und besteht ebenfalls aus einer Bezeichnung (S\_A3) und einer Beschreibung, die die Bedeutung des Skills für Lernende erläutern soll. Zudem erscheint es sinnvoll, einen *Skill* durch eine Grafik zu unterstützen. Deshalb soll, ähnlich wie bei einem *Bad-*

ge, die Möglichkeit bestehen, einen Dateipfad zu einer Grafik anzugeben, die einen *Skill* grafisch repräsentiert. *Skills* werden ebenso wie *Badges* freigeschaltet (siehe S\_A1), modelliert durch die konkrete *SkillBelohnungskonsequenz SkillFreischalten*.

#### Punktesystem:

Das GDE *Point* wird als ein System in der Klasse *Punktesystem* umgesetzt (siehe P\_A1). Es handelt sich um ein System, da Punkte nicht nur als einmalige Belohnung innerhalb eines spielifizierten Lehr-Lernarrangements auftauchen, sondern sich an verschiedenen Stellen wiederfinden. Dass es sich beim Modellieren immer um dasselbe *Punktesystem*-Objekt handeln muss, wenn dies an mehreren Stellen im Modell verwendet werden soll, ist eine Notwendigkeit, die jedoch nicht als Teil der Syntaxdefinition der DSML umgesetzt wird. Dabei handelt es sich vielmehr um eine Korrektheits- bzw. Konsistenzregel, die in die funktionale Verantwortlichkeit des Emendo Designers fällt. Das GDE bietet in der DSML ebenfalls die Möglichkeit, in dem Attribut *dateipfadZuGrafik* eine URL zu einem Icon anzugeben, welches Punkte repräsentiert und trägt damit zusammen mit dem Attribut *punkteSystemBezeichnung* zu der Umsetzung von Anforderung P\_A4 bei (der Punktwert selbst ist ein dynamischer Wert und ist deshalb nicht Teil der Sprache, sondern des LMS). Standardmäßig ist im letzteren Attribut der String Erfahrungspunkte gesetzt. Dieser kann, je nach Bedarf, beliebig überschrieben werden. Jede *Belohnungskonsequenz* eines Punktesystems geht von einem Punktwert aus, den es durch die drei verschiedenen Manipulationsarten zu verändern gilt (siehe P\_A5). *PunktesystemPunktestandErhoehen* erhöht den aktuellen *punktwert* um eine Ganzzahl, während *PunktesystemPunktestandVerringern* diesen um eine Ganzzahl verringert und schließlich *PunktesystemPunktestandSetzen* den *punktwert* durch einen neuen Ganzzahlwert ersetzt.

#### Levelsystem:

Bei einem *Levelsystem* greift die selbe Argumentation, wie bei dem *Punktesystem* in Bezug auf die Repräsentationsform durch ein System, bestehend aus einzelnen Leveln (siehe L\_A4). Ebenso kann hier eine für das gesamte *Levelsystem* gültige Grafik angegeben werden, die Anforderung L\_A2 mit umsetzt. Zudem erlaubt das *Levelsystem* in Bezug auf L\_A2 die Festlegung einer textuellen Bezeichnung durch das Attribut *levelSystemBezeichnung*. Außerdem können einzelne Level in *levelBezeichnungen* spezifiziert werden, wobei sich die Anzahl der unterschiedlichen Level durch die Angabe von einer Ober- bzw. Untergrenze ergibt (Attribute *max\_level* und *min\_level*). Ein *levelFaktor* sorgt

dafür, dass die Zuordnung von Levelbezeichnungen zu quantitativen Levelstufen nicht eine Reihe sein muss, die sich standardmäßig mit dem Faktor 1 (0, 1, 2, 3...) entwickelt, sondern hierbei auch weitere ganzzahlige Faktoren (z.B. für `levelFaktor = 10` -> 0, 10, 20, 30...) angegeben werden können. Einzelne Level innerhalb eines *Levelsystems* werden stufenweise durch die Klasse *LevelsystemStufeInkrementieren* inkrementiert und damit progressiv freigeschaltet (`L_A1`) bzw. durch *LevelsystemStufeDekrementieren* dekrementiert.

Durch die vorgestellte Art und Weise, dynamisches Verhalten in der DSML zu beschreiben, ist es möglich diverse strukturelle Anforderungen umzusetzen, die Abhängigkeiten von Game Design-Elementen als Belohnung für das Ausführen von Lernaktivitäten auf Lernelemente umsetzen. Durch die Einführung des Konzepts der Belohnungsregeln und ihren Bestandteilen, können damit die folgenden, sich auf die Belohnung von Lernenden beziehenden, strukturellen Anforderungen umgesetzt werden: `B_Z3`, `B_Z4`, `B_Z5`, `B_Z6`, `P_Z1`, `P_Z3`, `P_Z4`, `P_Z5`, `L_Z1`, `L_Z2`, `L_Z3`, `S_Z1`, `A_Z1`, `V_Z2`, `PC_Z2`.

Einen vergleichbaren Gestaltungsspielraum bieten Feedbackregeln, um individuell Feedback zu Lernaktivitäten, wie das Beantworten einer Frage (siehe `A_A4`) zu geben.

## 5.6 KONKRETE SYNTAX

Für die Entwicklung der konkreten Syntax (Notation) wurde den Empfehlungen von Kelly u. Tolvanen gefolgt und die potentiellen zukünftigen Nutzer der Emendo DSML mit einbezogen. Dies soll eine „not-invented-here“ Einstellung (Kelly u. Tolvanen, 2008, S. 257), also die Skepsis und teils damit verbundene Ablehnung von potentiellen Nutzern gegenüber Neueinführungen bzw. Neuentwicklungen (in diesem Fall der Emendo DSML) vermeiden. Ein frühzeitiges Einbeziehen der Nutzer in die Entwicklung der Notation hat neben der höheren Identifikation mit der Emendo DSML den Vorteil, dass die Verständlichkeit, Usability und Produktivität der DSML maßgeblich gesteigert werden kann (Frank, 2013, S. 150).

### 5.6.1 Nutzerstudie

Um eine möglichst fundierte konkrete Syntax zu erstellen, die auf verschiedenen Ideen von potentiellen Nutzern aufbaut und diese Ideen in Form eines kleinsten gemeinsamen Nenners umsetzt, wurde eine Nutzerbefragung (n=12) durchgeführt. Die Befragung hatte das Ziel, zu jedem einzelnen Domä-

nenkonzept der DSML einen Notationsvorschlag von Nutzern zu erhalten. Dafür wurde eine schriftliche Befragung in Form eines strukturierten Fragebogens<sup>25</sup> an potentielle Nutzer ausgehändigt. Darin waren die Teilnehmenden aufgefordert, einen Vorschlag für jeweils eine Notation zu vorgegebenen Domänenkonzepten zu skizzieren. Die Teilnahme an dieser Studie war freiwillig und es wurden Dozierende aus den Fachbereichen Informatik und Pädagogik für die Teilnahme gewonnen.

Dieser Input wurde in einem nächsten Schritt ausgehend von jedem Domänenkonzept gesichtet und daraufhin jeweils in Notationselemente umgesetzt. Dabei wurden die von Frank (2013, S. 15off.) vorgeschlagenen Guidelines<sup>26</sup> zum Design von Notationen berücksichtigt.

### 5.6.2 Umsetzung der konkreten Syntax

Bei der Betrachtung der Befragungsergebnisse zeigte sich, dass die zu Beginn der Studie erhoffte Vergleichbarkeit in der Darstellung der Domänenkonzepte eingetreten ist. Diese Vergleichbarkeit lässt sich exemplarisch an Abbildung 47<sup>27</sup> verdeutlichen. Die Notationselemente (1) und (2) sind Vorschläge für das Domänenkonzept Single Choice-Aufgabe aus der Nutzerbefragung, während (3) das daraus resultierende und aus den Nutzervorschlägen abgeleitete Notationselement darstellt.

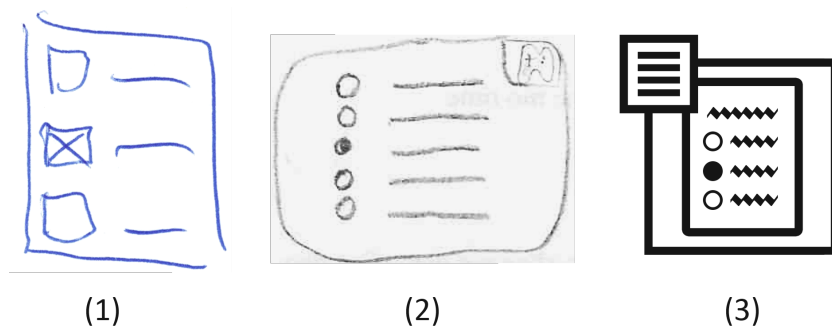


Abbildung 47: Entwicklung der Notationselemente der Emendo DSML am Beispiel einer Single Choice-Aufgabe

Bei den einzelnen Vorschlägen gab es sowohl grafisch minimalistische als auch umfangreiche Varianten. Diese Varianz wurde dadurch abgefangen, dass ein generisches Symbol für jede Kategorie eingeführt wurde, welches sich bei

<sup>25</sup> Der eingesetzte Fragebogen befindet sich im Anhang A.3 dieser Arbeit.

<sup>26</sup> Frank nennt insgesamt neun Guidelines (siehe Frank (2013, S. 15off.)), wie Notationen erstellt und gestaltet werden können.

<sup>27</sup> Bereits in ähnlicher Darstellung veröffentlicht in Bartel et al. (2017b, S. 1504).

den einzelnen Notationselementen einer Kategorie wiederfindet. Dieses Vorgehen entspricht der zweiten Guideline von Frank (2013, S. 151). Das generische Symbol ist beispielsweise bei Notationselementen von unterschiedlichen Aufgabentypen, Lernmaterialien sowie GDEs erkennbar (siehe Abbildung 48 (3) - Kasten an der linken oberen Ecke des Icons). Die erste Guideline nach Frank (2013) – die Aufteilung der Domänenkonzepte in semantische Kategorien – wurde implizit bereits durch die Struktur der abstrakten Syntax geleistet.

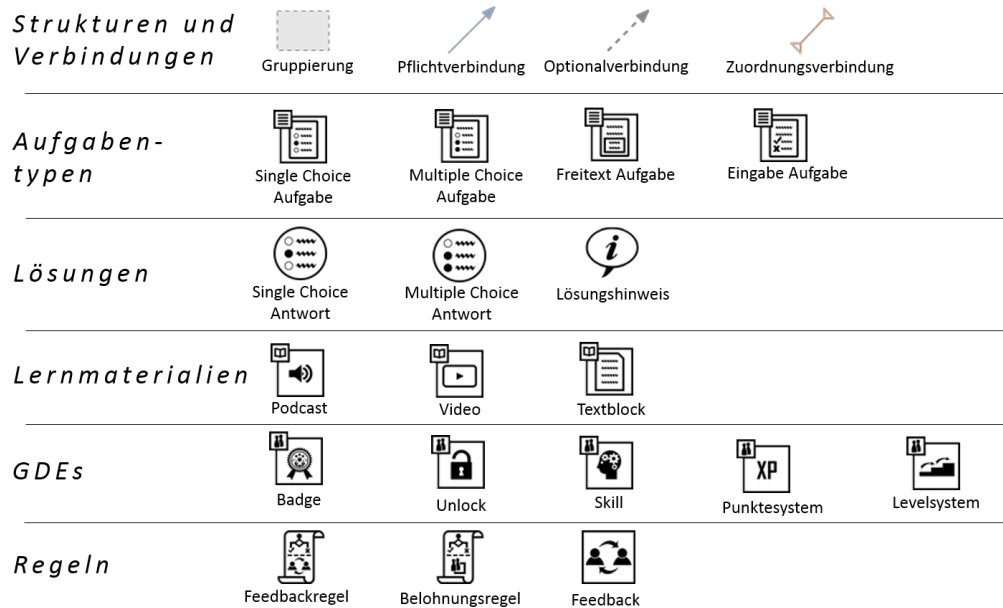


Abbildung 48: Konkrete Syntax der Emendo DSML

Die dritte Guideline sieht vor, dass je größer die semantische Distanz zweier Domänenkonzepte, desto verschiedener sollte die Notation dieser Konzepte sein (Frank, 2013, S. 151). Die Umsetzung dieser Guideline resultiert in der Verwendung von möglichst unterschiedlichen Icons, um die visuelle Distanz (Moody, 2009) zwischen den Notationselementen (sowohl innerhalb einer semantischen Kategorie, als auch von außerhalb) zu erhöhen. In der vierten Guideline schlägt Frank vor, Icons geometrischen Formen vorzuziehen, da Icons in der Lage sind, an gedankliche Modelle bei Nutzern von Konzepten einer Domäne besser anzuknüpfen (Frank, 2013, S. 151). Deshalb wurden bei der Notation bewusst konnotative Elemente verwendet (die allesamt aus Vorschlägen der Befragung stammen), wie beispielsweise Spielfiguren als generisches Symbol, was eine visuelle Verknüpfung zu Game Design-Elementen unterstützt. In Guideline Nummer fünf spricht sich Frank für die effektive Kombination von „[...] shapes (including icons), color and text [...]“ (Frank, 2013, S. 152) aus. Infolgedessen wurde bei dem Design der Notation für jedes Domänenkonzept ein Icon vorgesehen, welches in Kombination mit darunter-

liegendem Text der Form <Bezeichnung>\_<ID> für eine eindeutige Identifikation eines Objekts sorgen soll. Zudem wurde weitestgehend auf einen Einsatz von Farben aufgrund ihres nicht eindeutig positiven Charakters<sup>28</sup> verzichtet (Frank, 2013). Guideline sechs und sieben sehen vor, Notationselemente jeweilig immer zu genau einem Domänenkonzept zuzuordnen, sowie dabei darauf zu achten, dass Domänenkonzepte auch nur durch genau ein Notationselement in der Sprache repräsentiert sind (Frank, 2013, S. 152), was beides in der Emendo DSML eingehalten wurde. Die achte Guideline besagt, dass gleiche semantische Eigenschaften eines Domänenkonzepts durch eine Komposition von Notationen repräsentiert sein sollen (Frank, 2013, S. 152). Dieses Prinzip wird deutlich, wenn man sich beispielsweise die semantische Zusammengehörigkeit von einer Feedbackregel zu einem Feedback vor Augen führt. Eine Feedbackregel besteht auf semantischer Ebene aus mindestens einem Feedback-Objekt in der Rolle einer Feedbackkonsequenz. Diese Teil-Ganzes-Beziehung spiegelt sich auch in der Notation der Elemente wieder, vor allem in der der Feedbackregel. Das Notationselement einer Feedbackregel ist zweiteilig aufgebaut (siehe Abbildung 48): Der obere Teil des Elements konnotiert mit einer Regel, in dem er einen Entscheidungsbaum darstellt, während der untere Teil, das Notationselement eines Feedbacks aufgreift und damit die semantische Beziehung auch grafisch im Notationselement verdeutlicht. Selbiges gilt beispielhaft für die Notation einer Belohnungsregel, die das generische Symbol<sup>29</sup> einer Belohnungskonsequenz in Form eines GDE vereint. In der neunten und letzten Guideline spricht sich Frank für einen Einsatz von Notationselementen aus, welche visuelle Aggregationen abbilden, für den Fall, dass das Modellieren auf mehr als einer Abstraktionsstufe vorgesehen ist. Da dies in der Emendo DSML zwar möglich ist, aber nicht durch Notationselemente gelöst ist, wird diese Guideline zwar Beachtung geschenkt, jedoch aufgrund der Struktur der Emendo DSML, so nicht umgesetzt.

<sup>28</sup> Farben können sowohl grafisch diskriminierend wirken, als auch unterstützend, beispielsweise bei der impliziten Zusammengehörigkeit von Domänenkonzepten. Die Grundlage hierfür findet sich in psychologischen Gestehtgesetzen, vor allem in dem darin beschriebenen Gesetz der Ähnlichkeit, wonach ähnlich aussehende Objekte als zusammengehörig wahrgenommen werden (siehe hierzu z. B. Heimann u. Schütz (2017, S. 208f.)).

<sup>29</sup> Ein generisches Symbol ist bei dem Notationselement Feedback nicht notwendig, da es sich um genau ein Objekt dieser semantischen Kategorie handelt. Wird in der Weiterentwicklung der Sprache mehr als ein Feedback vorgesehen, ist die Einführung eines generischen Symbols, ähnlich wie bei Game Design-Elementen, sinnvoll.

## 5.7 ZWISCHENFAZIT

Dieses Kapitel geht auf die Grundlagen von DSM ein und stellt die Vorüberlegungen und die Entwicklung der Emendo DSML als einen wesentlichen Bestandteil des DSM-Ansatzes vor. Mit der DSML können statische Elemente eines spielifizierten Lehr-Lernarrangements, sowie dazugehöriges dynamisches Verhalten beschrieben werden. Gerade die dafür eingesetzten dynamischen Elemente (z. B. Feedback- oder Belohnungsregeln) bieten dabei ein hohes Maß an Flexibilität in der Gestaltung derartiger Lehr-Lernarrangements. Durch diese flexible Form der Definition solcher Arrangements mit den jeweiligen Ausprägungen und Bestandteilen ist es möglich, vielfältige Kombinationen aus Domänenobjekten zu schaffen, die jedoch nicht alle gemäß den Domänenregeln valide wären. So sollte beispielsweise eine für *Badges* spezifische *Belohnungskonsequenz* nicht in Bezug auf ein *Levelsystem* eingesetzt werden dürfen. Da dies laut den vorgestellten Klassendiagrammen möglich wäre, müssen Mechanismen existieren, welche diverse Objektkombinationen einschränken, idealerweise im Hintergrund während des Modellierens. Dies hat zur Folge, dass die Emendo DSML in einer Ausführungsumgebung eingebettet sein soll, die es nicht nur erlaubt, Modelle zu erstellen, die auf der abstrakten und konkreten Syntax der DSML basieren, sondern vielmehr die Korrektheit und Konsistenz dieser Modelle sicherstellen soll. Das Framework *Eclipse Sirius* erlaubt, auf Basis einer DSML, wie der vorliegenden, Modelle zu erstellen und zudem die hierfür notwendigen Korrektheits- und Konsistenzregeln zu definieren. Sirius basiert dabei auf EMF, was als Grundlage für die Verarbeitung von Domänenobjekten durch eine Anwendungsebene betrachtet werden kann. Im Folgenden werden diese Aspekte diskutiert und damit der siebte und letzte Schritt in der Definition einer DSM-Lösung – die Umsetzung der Werkzeugunterstützung – beschrieben. In den einzelnen Werkzeugen der Emendo Werkzeuglandschaft werden zudem weitere strukturelle Anforderungen umgesetzt, die nicht direkt in die DSML eingeflossen sind.



---

EMENDO DESIGNER UND GENERATOR

---

*„Perfection (in design) is achieved not when  
there is nothing more to add,  
but rather when there is nothing more  
to take away.“*

*— Antoine de Saint-Exupery*

## 6.1 EINLEITUNG

Das vorherige Kapitel 5 stellte die Grundlagen des DSM-Ansatzes vor sowie die Umsetzung der Emendo DSML anhand der zuvor erhobenen konzeptuellen Anforderungen. Die abstrakte und konkrete Syntax der Emendo DSML sind vollständig definiert. Jedoch kann in diesem Stadium keine Weiterverarbeitung von abstrakter und konkreter Syntax im Kontext des DSM-Ansatzes stattfinden. Nach dem Ansatz wäre der nächste Schritt, Modelle auf Basis der Definitionen der DSML erstellen zu können. Es bedarf einer Übertragung beider Syntaxtypen in jeweils ausführbare bzw. nutzbare Formate, auf Basis derer Modelle durch Nutzer erstellt werden können. Hierfür stellt das vorliegende Kapitel die fachlichen Grundlagen für die jeweilig ausgewählten Technologien dar und veranschaulicht, wie die ursprüngliche Definition der Emendo DSML in Form von Klassendiagrammen und grafischen Notationselementen sukzessive in eine für Nutzer anwendbare DSML überführt wird, mit der valide Modelle von spielifizierten Lehr-Lernarrangements erstellt werden können. Aus Gründen der Nachvollziehbarkeit und Transparenz sollen die einzelnen Schritte beispielhaft anhand der Klassen *Hinweis* und *Aufgabe*, die im Metamodell der Emendo DSML definiert sind, veranschaulicht werden. Die Abbildung 49 zeigt diesen Ausschnitt des Metamodells.

Analog zu den beiden beispielhaft herangezogenen Klassen sind alle weiteren Klassen des Metamodells entsprechend umgesetzt worden.

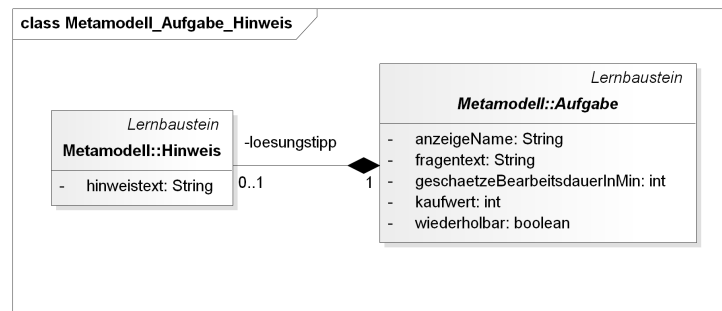


Abbildung 49: Auszug aus Emendo DSML Metamodell - Beziehung von Hinweis zu Aufgabe

## 6.2 ECLIPSE MODELING FRAMEWORK

Für die Umsetzung der abstrakten Syntax von UML-Klassendiagrammen in eine durch Software verarbeitbare Meta-Sprache wurde das auf Eclipse basierende Modellier-Framework EMF verwendet. EMF erlaubt die Erstellung von Tools, die auf einem strukturierten Datenmodell basieren. Dieses Datenmodell kann in Form von UML-Modellen, XML-Schema, Relationale Datenbank (RDB)-Schema oder weiteren Eingabeformaten definiert werden (siehe Abbildung 50). Das Datenmodell in EMF wird verwendet, um daraus nach dessen Definition auf Java-basierende und annotierte Implementierungs- und Interfaceklassen zu generieren, die Services bereitstellen, um Modelle zu Manipulieren, sie in XMI zu serialisieren oder zu laden oder Änderungen am Modell nachzuverfolgen (Combemale et al., 2017, S. 56).

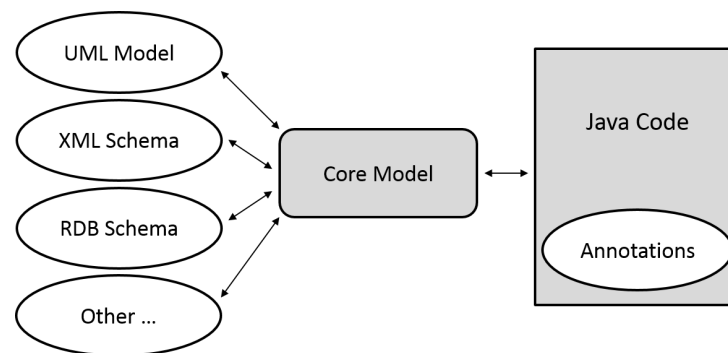


Abbildung 50: Ein- und Ausgabe von Ecore (Steinberg et al., 2009, S. 21)

### 6.2.1 Ecore und seine Sprachkonzepte

Um derartige Datenmodelle in EMF abzubilden, wird ein Ecore-Modell verwendet. Ecore stellt die Meta-Sprache von EMF dar, da Ecore selbst ein Modell

von EMF ist, und kann deshalb als eine Umsetzung von MOF angesehen werden (Steinberg et al., 2009, S. 15; Combemale et al., 2017, S. 54). Diese Vergleichbarkeit wird deutlich, wenn man sich die wesentlichen Bestandteile von Ecore und deren Abhängigkeiten untereinander verdeutlicht. Abbildung 51 zeigt eine Übersicht über die wesentlichen Konzepte in Ecore in einem UML Klassendiagramm.

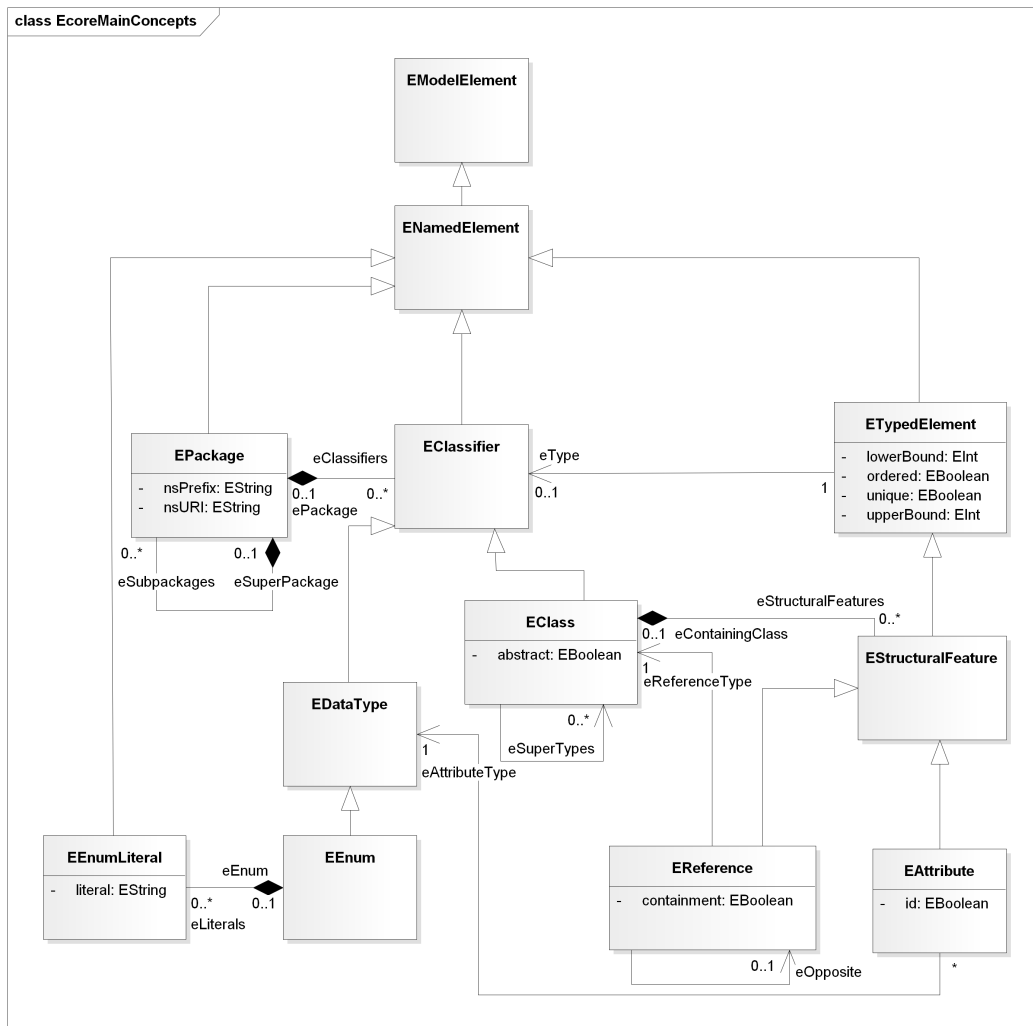


Abbildung 51: Überblick über wesentliche Sprachkonzepte von Ecore nach (Brambilla et al., 2017, S. 101)

Die dabei wesentlichen Bestandteile, welche häufig als vereinfachtes Metamodell von Ecore beschrieben werden (siehe z. B. (Steinberg et al., 2009, S. 16)), sind *EClass*, *EAttribute*, *EReference* und *EDataType* (Steinberg et al., 2009, S. 16; Brambilla et al., 2017, S. 100f.; Combemale et al., 2017, S. 54f.):

- *EClass* wird verwendet, um eine modellierte Klasse zu repräsentieren und ist damit das Äquivalent zu einer *Class* in MOF. Jede *EClass* besitzt einen Namen, kann abstrakt sein, Elternklassen (*eSuperTypes*) besitzen

und diverse *EStructuralFeatures* haben, die entweder vom Typ *EAttribute* (Attribute in MOF) oder *EReference* (Referenzattribute zwischen *EClasses*) sein können.

- *EAttribute* repräsentieren modellierte Attribute und besitzen neben einem Namen (abgeleitet von *ENamedElement*) und einem Datentyp *EDataType* Attribute der Elternklasse *ETypedElement*, um beispielsweise die Multiplizitäten anzugeben, in dem *lowerBound* und *upperBound* entsprechen überschrieben werden.
- *EReference* wird benutzt, um ein Ende einer Assoziation zwischen *EClasses* zu repräsentieren. Dabei kann neben einem Namen und dem Referenztyp angegeben werden, ob es sich um eine *containment*-Beziehung handelt, also eine Komposition in der Analogie der UML vorliegt. Bei bidirektionalen Beziehungen werden demnach zwei *EReferences* benötigt, was in einer *EReference* durch die Belegung der *eOpposite* definiert werden kann. Wie auch bei *EAttributes* können bei *EReferences* Attribute spezifiziert werden, die aus *ETypedElement* abgeleitet werden und die Verbindung beispielsweise in ihrer Multiplizität einschränken.
- *EDataTypes* wird verwendet, um primitive Typen von Attributen (z. B. *EString*, *EBoolean*, *EInt*, *EFloat*) zu repräsentieren.
- *EEnums* sind primitive Typen (*EDataType*), die dem Konzept von *Enumeration* in MOF entsprechen und Literale (*EEnumLiteral*) in Form von *EStrings* enthalten.
- *EPackages* gruppieren *EClassifiers* und können ein Elternpaket (*eSuperPackage*) besitzen oder Kindpakete darstellen (*eSubpackages*). Um *EPackages* eindeutig zu adressieren, kann ein *nsURI* angegeben und ein *nsPrefix* festgelegt werden, was als Prefix in der Bezeichnung der zu generierenden Java-Klassen erscheinen soll.

### 6.2.2 Umsetzung der abstrakten Syntax in Ecore

Die Übertragung nach Ecore des Emendo UML-Metamodells erfolgte manuell, nachdem das Metamodell eine entsprechende inhaltliche Stabilität aufwies, was nach mehreren Iterationen und daraus entstehenden Inkrementen der Fall war. Beispielhaft soll im Folgenden die Umsetzung der Metaklasse *Hinweis* in Ecore beschrieben werden:

Die Darstellung der Metaklasse *Hinweis* erfolgt mit der Syntax von Ecore und wurde, wie in dem Code-Auszug 4 dargestellt, umgesetzt. Zu erkennen ist

neben der Deklaration der Klasse *Hinweis* durch einen *eClassifiers*-Tag (Zeile 1) auch die Deklaration eines Attributs der Klasse *Hinweis* in Form eines *eStructuralFeatures* (Zeile 2) mit den entsprechenden Attributen der Tags, die bereits aus Abbildung 51 bekannt sind.

Code 4: Darstellung der Klasse Hinweis in Ecore

```

1 <eClassifiers xsi:type="ecore:EClass" name="Hinweis" eSuperTypes="#//
  Lernbaustein">
2   <eStructuralFeatures xsi:type="ecore:EAttribute" name="hinweistext"
     unique="false" eType="ecore:EDataType http://www.eclipse.org/emf
     /2002/Ecore#//EString"/>
3 </eClassifiers>

```

Analog zur der Ecore-Repräsentation der Klasse Hinweis, wird die abstrakte Klasse Aufgabe exemplarisch in Code-Auszug 5 dargestellt. Erkennbar ist zudem, dass die im Metamodell beschriebene Komposition durch eine *containment*-Beziehung (siehe Zeile 2) in Ecore umgesetzt wird, die in Ecore nur in Aufgabe vorkommt, nicht aber in Hinweis und damit die bereits beschriebene Handhabung von UML-Beziehungstypen in Ecore praktisch aufzeigt.

Code 5: Hinweis als Referenz in Aufgabe in Ecore

```

1 <eClassifiers xsi:type="ecore:EClass" name="Aufgabe" abstract="true"
  eSuperTypes="#//Lernbaustein">
2   <eStructuralFeatures xsi:type="ecore:EReference" name="loesungstipp"
     unique="false" eType="#//Hinweis" containment="true"/>
3 </eClassifiers>

```

### 6.2.3 Generierung von Eclipse Plug-In-Projekten

Wie bereits erwähnt und veranschaulicht, ist XMI<sup>1</sup> das Format, in welchem sich Ecore-Modelle definieren lassen. Dies erlaubt eine Serialisierung, ohne zusätzliche Informationen, wie sie beispielsweise in annotierten Java-Klassen als Eingabedaten für Ecore enthalten sind. Folglich ist jedes Ecore-Modell durch eine einheitliche Syntax definiert, was eine einheitliche Behandlung von Modellen bedingt. Deshalb ist es möglich, aus Ecore Modellen, automatisiert die bereits beschriebenen Services für die Manipulation von Modellen, deren Serialisierung sowie einen Event-Subscriber-Mechanismus zu generieren. Dies geschieht mit Hilfe eines weiteren *EMF*-Modells, des Generator-Modells (*genmodel*), was neben den Informationen aus dem Ecore-Modell, noch weitere Informationen enthält, wie beispielsweise in welchem Verzeich-

<sup>1</sup> Die OMG-Spezifikation von XMI ist zu finden unter: <http://www.omg.org/spec/XMI>

nis der zu generierende Code abgelegt werden soll. Das Prinzip ist vergleichbar mit dem Dekorierer-Pattern (Gamma et al., 1995, S. 196ff.), da Generator Modell-Klassen Dekorierer für Ecore-Klassen darstellen (Steinberg et al., 2009, S. 26). Wird die Generierung durch den EMF-internen Generator mit einem *genmodel* ausgeführt, so werden vier Eclipse-Plug-In-Projekte erstellt (Combemale et al., 2017, S. 57):

- Ein *model*-Projekt, welches die Implementierung des Metamodells enthält, wobei für jede Klasse im Metamodell ein Interface und eine dazugehörige annotierte Implementierungsklasse generiert wird. Der Code-Ausschnitt 6 zeigt die generierte Interface-Klasse aus der Ecore bzw. *genmodel*-Repräsentation der Metaklasse *Hinweis*. Zu erkennen sind die einzelnen Annotationen in Form von Kommentaren, welche jedoch geändert werden können. Das Hinzufügen des Schlüsselworts *not* nach der *@generated*-Annotation (siehe Code-Ausschnitt 7) bewirkt beispielsweise, dass Änderungen, die innerhalb einer Klasse oder Methode manuell getroffen wurden, bei einem erneuten Ausführen des *genmodels* nicht überschrieben werden. Dies ist insofern sinnvoll, als dass EMF das Setzen von Standardwerten für vererbte Attribute polymorph nicht unterstützt, was jedoch manuell im Code nachbearbeitet werden kann. Dies wurde beispielsweise für das Setzen der Standardwerte für den jeweilig Klassen-abhängigen Variablennamen und das jeweilige Variablenanzeigetemplate, stammend aus dem bereits beschriebenen Variablenkonzept der abstrakten Syntax der Emendo DSML verwendet.

Code 6: Generiertes Java Interface zur Ecore-Repräsentation der Klasse Hinweis

```
1 package org.eclipse.emendo;
2
3 /**
4  * @model
5  * @generated
6  */
7 public interface Hinweis extends Lernbaustein {
8     /**
9      * @model unique="false"
10     * @generated
11     */
12     String getHinweistext();
13
14     /**
15     * @generated
16     */
17     void setHinweistext(String value);
```

```

18
19 } // Hinweis

```

Die dazugehörige Implementierungsklasse ist in Code-Ausschnitt 7 abgebildet. Man erkennt darin, neben den Standardzugriffsmethoden, ebenso Methoden, die EMF-spezifisch diese verwenden (*eSet(...)*, *eUnset(...)*) sowie eine Methode zur Überprüfung, ob ein Feature, identifiziert durch eine *featureID* gesetzt ist (*eIsSet(...)*). In den Standardzugriffsmethoden ist ein Benachrichtigungsmechanismus gemäß dem Beobachter-Muster (Gamma et al., 1995, S. 326ff.) umgesetzt, womit andere Klassen sich als Beobachter für einen Status eines Objekts der Klasse registrieren können und benachrichtigt werden, sofern sich dieser ändert (Zeile 46f.). Darüber hinaus kann jede Klasse und deren Features statisch jeweils über eine ID in *EmendoPackage* identifiziert werden (siehe *eStaticClass()*), was einen direkten Zugriff auf die jeweiligen Objekte oder deren Attribute erlaubt, um beispielsweise Attributwerte zu ändern oder ein Objekt einer Klasse über eine von EMF bereitgestellte Factory, gemäß dem Factory-Method-Pattern (Gamma et al., 1995, S. 121ff.) zu erzeugen.

Code 7: Generierte Java Interface-Implementierungsklasse zur Ecore-Repräsentation der Klasse Hinweis

```

1 package org.eclipse.emendo.impl;
2
3 // imports, die aus Gruenden der Uebersichtlichkeit entfernt
  wurden
4
5 /**
6  * @generated
7  */
8 public class HinweisImpl extends LernbausteinImpl implements
   Hinweis {
9     /**
10     * @generated
11     * @ordered
12     */
13     protected static final String HINWEISTEXT_EDEFAULT = null;
14
15     /**
16     * @generated
17     * @ordered
18     */
19     protected String hinweistext = HINWEISTEXT_EDEFAULT;
20
21     /**
22     * @generated not
23     */
24     public HinweisImpl() {

```

```

25     super();
26 }
27
28 /**
29  * @generated
30  */
31 @Override
32 protected EClass eStaticClass() {
33     return EmendoPackage.Literals.HINWEIS;
34 }
35
36 /**
37  * @generated
38  */
39 public String getHinweistext() {
40     return hinweistext;
41 }
42
43 /**
44  * @generated
45  */
46 public void setHinweistext(String newHinweistext) {
47     String oldHinweistext = hinweistext;
48     hinweistext = newHinweistext;
49     if (eNotificationRequired())
50         eNotify(new ENotificationImpl(this, Notification.SET,
51             EmendoPackage.HINWEIS__HINWEISTEXT, oldHinweistext,
52             hinweistext));
53 }
54
55 /**
56  * @generated
57  */
58 @Override
59 public Object eGet(int featureID, boolean resolve, boolean
60     coreType) {
61     switch (featureID) {
62         case EmendoPackage.HINWEIS__HINWEISTEXT:
63             return getHinweistext();
64     }
65     return super.eGet(featureID, resolve, coreType);
66 }
67
68 /**
69  * @generated
70  */
71 @Override
72 public void eSet(int featureID, Object newValue) {
73     switch (featureID) {
74         case EmendoPackage.HINWEIS__HINWEISTEXT:
75         setHinweistext((String)newValue);
76         return;
77     }
78 }

```



```

75     super.eSet(featureID, newValue);
76 }
77
78 /**
79  * @generated
80  */
81 @Override
82 public void eUnset(int featureID) {
83     switch (featureID) {
84         case EmendoPackage.HINWEIS__HINWEISTEXT:
85             setHinweistext(HINWEISTEXT_EDEFAULT);
86             return;
87     }
88     super.eUnset(featureID);
89 }
90
91 /**
92  * @generated
93  */
94 @Override
95 public boolean eIsSet(int featureID) {
96     switch (featureID) {
97         case EmendoPackage.HINWEIS__HINWEISTEXT:
98             return HINWEISTEXT_EDEFAULT == null ? hinweistext != null : !
                HINWEISTEXT_EDEFAULT.equals(hinweistext);
99     }
100     return super.eIsSet(featureID);
101 }
102
103 /**
104  * @generated
105  */
106 @Override
107 public String toString() {
108     if (eIsProxy()) return super.toString();
109
110     StringBuffer result = new StringBuffer(super.toString());
111     result.append(" (hinweistext: ");
112     result.append(hinweistext);
113     result.append(')');
114     return result.toString();
115 }
116 }

```

- Ein *edit*-Projekt, dass als Brücke zwischen dem Eclipse User Interface (welches auf JFace basiert) und dem EMF Core fungiert und auf Grundlage des Command Patterns (Gamma et al., 1995, S. 263ff.) Funktionalität für Eclipse Commands (z. B. *Content- und Itemproviders*, Zugriff auf *Properties* von EMF Objekten etc.) und Modifikationen (z. B. *SetCommand*, *AddCommand* oder *CopyCommand* (Steinberg et al., 2009, S. 59f.)) von

auf EMF-basierenden Modellen bereitstellt (Steinberg et al., 2009, S. 46). Code-Ausschnitt 8 zeigt einen derartigen *Itemprovider* für die Metaklasse *Hinweis*. Bei einem *Itemprovider* handelt es sich allgemein um einen Adapter (gemäß Adapter-Pattern (Gamma et al., 1995, S. 157ff.)) für EMF-Objekte, der alle Schnittstellen für das Anzeigen oder Editieren dieser Objekte bereitstellt (Steinberg et al., 2009, S. 46) und damit die folgenden Funktionalitäten erfüllt (Steinberg et al., 2009, S. 47): Implementierung von Funktionen für Content- und Labelprovider (siehe Zeile 52ff. bzw. 68ff.), Bereitstellung von *Property Descriptors*<sup>2</sup> (siehe Zeile 20ff. bzw. 32ff.), Bereitstellung einer Command Factory für Commands auf Objekten der EMF-Klassen, Weiterleitung von Veränderungsbenachrichtigungen von EMF-Objekten an die entsprechenden Viewer (siehe Zeile 90ff.).

Code 8: Generierter Java Itemprovider der Klasse Hinweis

```

1 package org.eclipse.emendo.provider;
2
3 // imports, die aus Gruenden der Uebersichtlichkeit entfernt
  wurden
4
5 /**
6  * @generated
7  */
8 public class HinweisItemProvider extends LernbausteinItemProvider
9 {
10     /**
11     * @generated
12     */
13     public HinweisItemProvider(AdapterFactory adapterFactory) {
14         super(adapterFactory);
15     }
16
17     /**
18     * @generated
19     */
20     @Override
21     public List<IItemPropertyDescriptor> getPropertyDescriptors(
22         Object object) {
23         if (itemPropertyDescriptors == null) {
24             super.getPropertyDescriptors(object);
25
26             addHinweistextPropertyDescriptor(object);
27         }
28         return itemPropertyDescriptors;
29     }
30 }

```

<sup>2</sup> *Property Descriptors* legen fest, wie ein Property (Feature) im Editor dargestellt wird (Steinberg et al., 2009, S. 50).

```

29  /**
30  * @generated
31  */
32  protected void addHinweistextPropertyDescriptor(Object object) {
33      itemPropertyDescriptors.add
34      (createItemPropertyDescriptor
35      (((ComposeableAdapterFactory)adapterFactory).
36          getRootAdapterFactory(),
37          getResourceLocator(),
38          getString("_UI_Hinweis_hinweistext_feature"),
39          getString("_UI_PropertyDescriptor_description", "
40              _UI_Hinweis_hinweistext_feature", "_UI_Hinweis_type"),
41          EmendoPackage.Literals.HINWEIS__HINWEISTEXT,
42          true,
43          false,
44          false,
45          ItemPropertyDescriptor.GENERIC_VALUE_IMAGE,
46          null,
47          null));
48  }
49  /**
50  * @generated
51  */
52  @Override
53  public Object getImage(Object object) {
54      return overlayImage(object, getResourceLocator().getImage("
55          full/obj16/Hinweis"));
56  }
57  /**
58  * @generated
59  */
60  @Override
61  protected boolean shouldComposeCreationImage() {
62      return true;
63  }
64  /**
65  * @generated
66  */
67  @Override
68  public String getText(Object object) {
69      Hinweis hinweis = (Hinweis)object;
70      return getString("_UI_Hinweis_type") + " " + hinweis.getId();
71  }
72
73
74  /**
75  * @generated
76  */
77  @Override
78  public void notifyChanged(Notification notification) {

```

```

79     updateChildren(notification);
80
81     switch (notification.getFeatureID(Hinweis.class)) {
82         case EmendoPackage.HINWEIS__HINWEISTEXT:
83             fireNotifyChanged(new ViewerNotification(notification,
84                 notification.getNotifier(), false, true));
85             return;
86     }
87     super.notifyChanged(notification);
88 }
89
90 /**
91  * @generated
92  */
93 @Override
94 protected void collectNewChildDescriptors(Collection<Object>
95     newChildDescriptors, Object object) {
96     super.collectNewChildDescriptors(newChildDescriptors, object);
97 }

```

- Ein *editor*-Plug-In-Projekt, welches einen grundlegenden Eclipse Editor in Form einer Baumansicht (engl. Treeview) implementiert, mit dem Modelle auf Basis des in Ecore-definierten Metamodells erstellt werden können. Die dafür automatisch durch EMF generierten Java Klassen definieren beispielsweise grundlegende Graphical User Interface (GUI)-Komponenten, wie die *Action Bar* und dafür notwendige Beobachter einer Eclipse Instanz oder einen Wizard zum Erstellen von Modellen, die auf dem Emendo EMF-Modell basieren.
- Ein *test*-Projekt, das erlaubt, die Implementierungen im *model* Projekt anhand von bereitgestellten JUnit-Tests zu testen und das die dafür notwendigen Testfälle definiert.

Mit den vorgestellten Eclipse Plug-In-Projekten kann eine neue Eclipse Instanz erzeugt werden (in der die Projekte als Plug-Ins eingesetzt werden und welche eine Referenz zu der ursprünglichen Emendo Ecore-Datei besitzt), mit der auf der EMF (Ecore) Definition der Emendo DSML basierende Modelle mit Hilfe eines Treeview-Editors erstellt werden können. Diese Form des Modellierens lässt jedoch den Einsatz der konkreten Syntax der Emendo DSML gänzlich unberücksichtigt. Es bedarf demnach einer Möglichkeit, welche die in diesem Stadium verarbeitbare abstrakte Syntax mit der konkreten Syntax verbindet und benutzerfreundlichere Formen der Modellierung erlaubt.

### 6.3 SIRIUS

Eine auf der Eclipse-Plattform basierende Möglichkeit, diese bidirektionale Verknüpfung von abstrakter und konkreter Syntax herzustellen, ist das *Eclipse Sirius Framework*<sup>3</sup> (Obeo GmbH, 2017b; Combemale et al., 2017, S. 111, 290; Brambilla et al., 2017, S. 105). Sirius wird verwendet, um „[...] multi-view modeling workbenches through graphical, table, or tree modeling editors“ (Combemale et al., 2017, S. 111), angepasst an eine DSML, zu entwickeln. Sirius reduziert die Entwicklungszeit, indem es den komplexen Prozess der Erstellung eines Editors unterstützt, und erhöht damit die gesamte Produktivität bei der Erstellung eines grafischen Editors für eine DSML erheblich (Viyović et al., 2014, S. 233f.).

#### 6.3.1 Systembestandteile

Technisch gesehen bettet sich Sirius auf oberster Ebene in eine hierarchisierte Systemarchitektur ein, die auf unterster Ebene EMF integriert (siehe Abbildung 52).

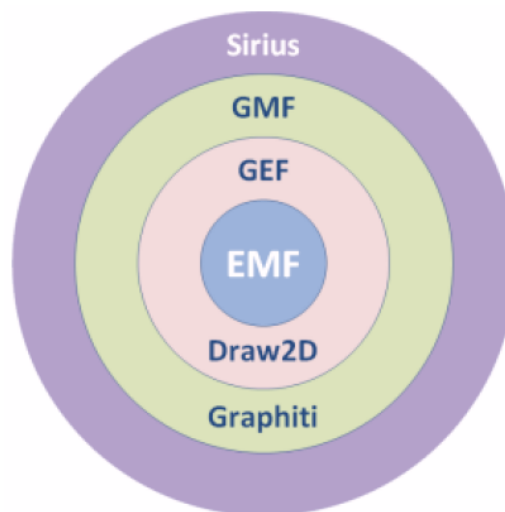


Abbildung 52: Hierarchische Darstellung der Systemarchitektur eines auf Eclipse basierenden Modellierungswerkzeugs (Viyović et al., 2014, S. 234)

Die darüber liegenden Frameworks, das Graphical Editor Framework (GEF) und Draw2D, stellen Grundfunktionalitäten bereit, um grafische Views auf Basis von EMF zu erstellen (Viyović et al., 2014, S. 234). Durch das Graphical Modeling Framework (GMF) werden die Frameworks GEF und Draw2D gekapselt. In Verbindung mit Graphiti, was die Komplexität von GEF in Form einer

<sup>3</sup> Version 4.0.

Fassade (Gamma et al., 1995, S. 208ff.) reduziert und ein Mapping zwischen EMF und GEF bereitstellt, ist es möglich, „[...] homogeneous graphical editors that visualize an underlying domain model based on a tool-defined graphical notation“ (Viyović et al., 2014, S. 234) zu designen.

### 6.3.2 Erstellung des Emendo Designers mit Sirius

Die Definition des Editors findet größtenteils nicht auf Code-Ebene statt, sondern vielmehr durch hierarchisierte sowie deklarative Beschreibungen in einem Viewpoint Specification Model (VSM), welches Struktur, Darstellung und Verhalten des Editors und der darin eingebetteten DSML festlegt (Combemale et al., 2017, S. 111; Viyović et al., 2014, S. 234). Konkret finden sich darin die folgenden Konzepte (siehe Abbildung 53, Markierungen 1–5) (Viyović et al., 2014, S. 235):

1. *Viewpoint*: Ist als Kern-Element eine logische Vereinigung aus Spezifikationen der Repräsentationen von Elementen sowie Code-basierten Erweiterungen dieser Spezifikationen.
2. *Representation*: Spezifikationen sind Gruppen aus grafischen Elementen, die Daten aus der abstrakten Syntax repräsentieren und ihre Struktur, ihre Darstellung und ihr Verhalten in Modellen in Repräsentationsformen von Diagrammen, Tabellen, Matrizen oder Bäumen darstellen.
3. *Mapping*: Mappings sind stark abhängig von Repräsentationsformen und definieren eine Teilmenge an semantischen Elementen, die in Repräsentationsformen dargestellt werden.
4. *Style*: Wird verwendet, um die visuelle Darstellung der Elemente festzulegen.
5. *Tool*: Beschreibt das Verhalten der Mappings.

Code in Form von AQL oder Java, wird nur an den Stellen im VSM eingesetzt, an denen die Standarddefinitionen durch Sirius nicht ausreichend sind. Dies kann beispielsweise bei der Festlegung von Korrektheits- und Konsistenzregeln (z. B. Abbildung 53, A) hilfreich sein, wenn die an OCL angelehnte und auf dem Model to Text Language (MTL)<sup>4</sup>-Standard basierte Alternative AQL nicht mehr ausreichend für die Formulierung komplexer Überprüfungen

<sup>4</sup> Siehe: <http://www.omg.org/spec/MOFM2T/1.0/>

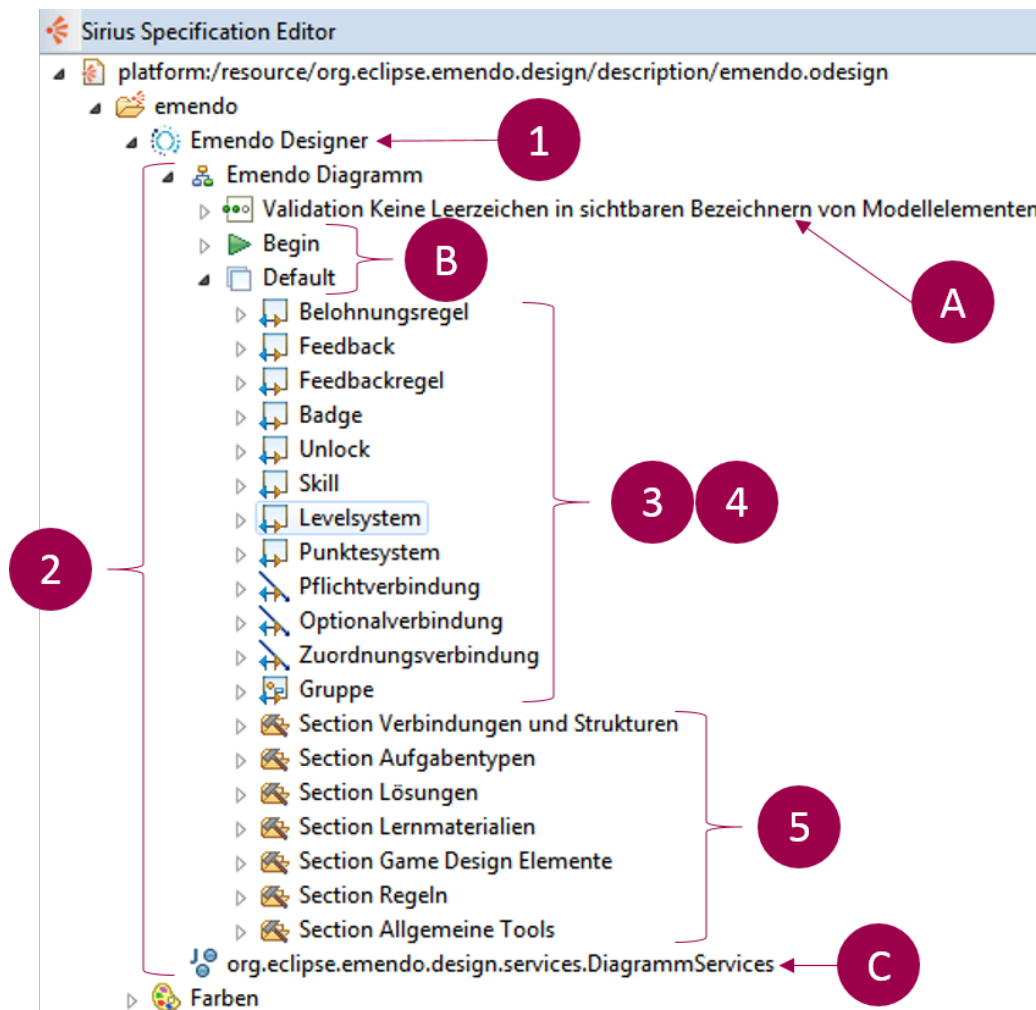


Abbildung 53: Struktur des Emendo DSML Viewpoint Specification Models

ist und demnach auf in Java verfasste Extensions (z. B. Abbildung 53, C) ausgewichen werden muss (Obeo GmbH, 2017b). Ebenso bietet Sirius die Möglichkeit Aktionen einzubetten, die nach Programmstart ausgeführt werden (z. B. Abbildung 53, B, *Begin*), oder ein Layerkonzept, bei dem die Definition von Diagrammbestandteilen zur Modellierung auf unterschiedlichen Abstraktionslayern möglich ist (z. B. Abbildung 53, B, *Default*) (Obeo GmbH, 2017b).

### 6.3.3 GUI-Überblick

Die Abbildung 54 zeigt eine Übersicht über den auf der Eclipse-Plattform basierenden Emendo Designer. Das Interface wurde visuell entschlackt (z. B. Ausblenden diverser *activities* und *perspectiveExtensions* aus den gleichnamigen Subpackages von *org.eclipse.ui*), um die Usability des Designers zu erhöhen und Benutzer nicht visuell zu überfordern.

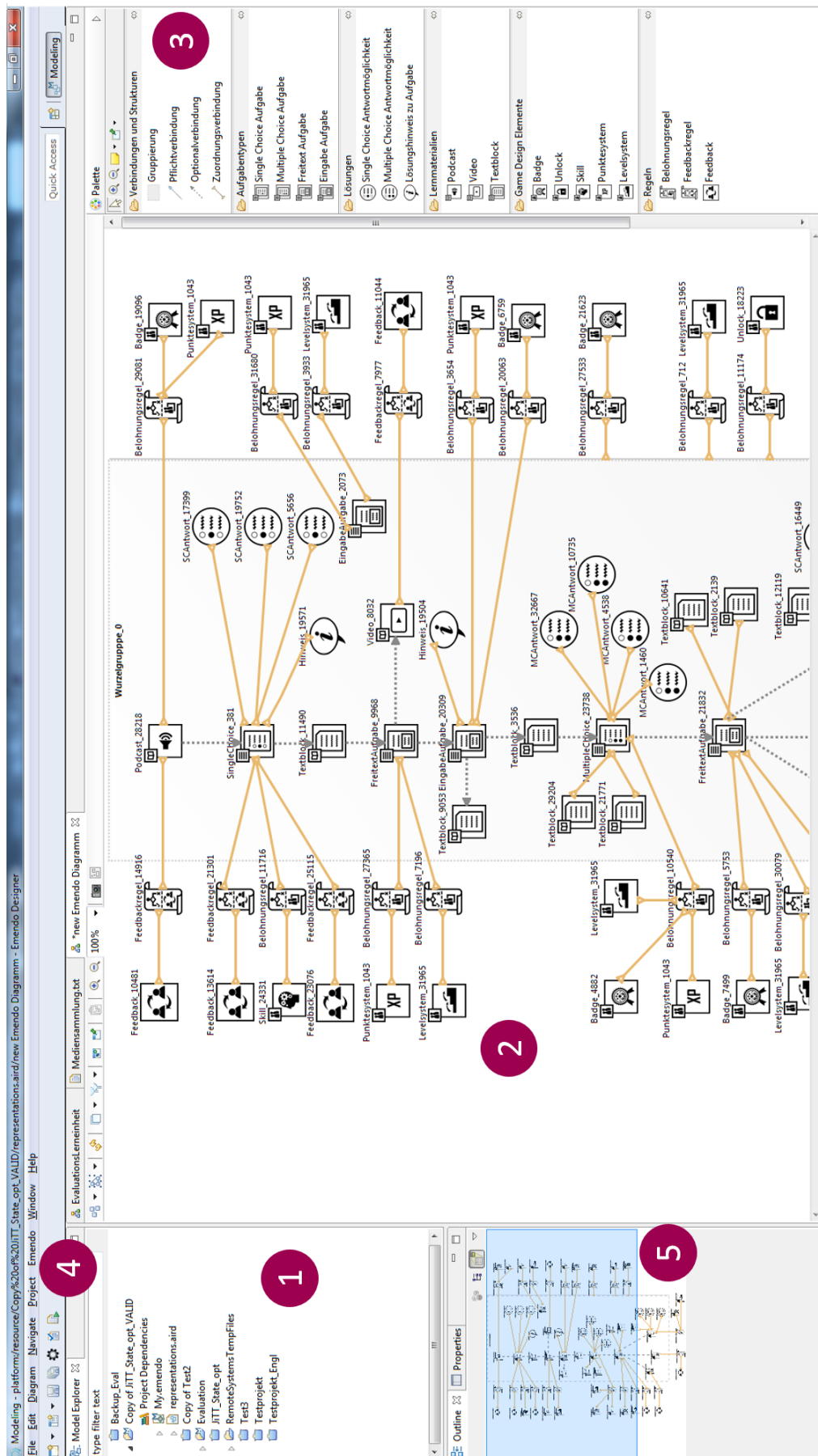


Abbildung 54: GUI des Emendo Designers



Die GUI des Designers setzt sich aus insgesamt fünf wesentlichen Bereichen zusammen, wie in Abbildung 54 (1–5) zu erkennen ist:

1. Der **Modell-Explorer** dient zur Verwaltung von Modellierungsprojekten.
2. Auf der **Zeichenfläche** können Notationselemente per Drag & Drop platziert und anschließend per Drag verschoben werden. Außerdem startet ein Doppelklick auf Notationselemente jeweilig einen JFace Dialog zur weiteren Eingabe von Informationen.
3. Die **Objektpalette** zeigt alle verfügbaren Notationselemente mit entsprechendem Icon, geordnet nach Elementtyp, die auf der Zeichenfläche platziert werden können.
4. Das **Menu** und die **Toolbar** wurden entsprechend entschlackt und bieten Shortcuts für die Validierung und Generierung sowie für die allgemeine Konfiguration (z. B. Festlegung der Lernziele, Angaben zu Ersteller und Kurzbeschreibung) der Lerneinheit.
5. Bei größeren Modellen bietet die **Outline** eine Gesamtübersicht in höherer Skalierung.

#### 6.4 RELEVANTE MODELLIERUNGSASPEKTE DES EMENDO DESIGNERS

Im Emendo Designer sind unterschiedliche Typen von Korrektheits- und Konsistenzüberprüfungen umgesetzt, welche zur Erstellung von validen Modellen auf Basis der abstrakten Syntax der Emendo DSML beitragen. Im Folgenden sollen Beispiele für die eingesetzten Techniken zur spezifischen Vermeidung von Problemen vorgestellt werden, mit denen das Ziel der Validität bei Modellen erreicht werden soll. Aus Gründen des Umfangs können nicht alle eingesetzten Maßnahmen hier diskutiert werden, sondern lediglich einzelne Repräsentanten.

##### 6.4.1 *Korrektheitsüberprüfungen bei Zuordnungsverbindungen*

Dieser Regeltypus legt fest, welche notwendige Bedingungen erfüllt sein müssen, um Modellierungsaktionen beim Erstellen von Zuordnungsverbindungen zuzulassen oder zu verweigern.

Beispielhaft kann hier die Aktion angeführt werden, bei der zwei Notationselemente mit einer Zuordnungsverbindung verbunden werden sollen. Ei-

ne Zuordnungsverbindung darf nur zwischen bestimmten Elementtypen existieren und wäre im Falle einer Verbindung eines Objekts vom Typ *GameDesignElement* mit einem Objekt vom Typ *Belohnungsregel* (oder vice-versa) durch die AQL-Ausdrücke in Codeausschnitt 9 abgesichert.

Code 9: AQL-Constraints zur Erstellung einer Zuordnungsverbindung zwischen zwei Notationselementen

```

1 aql:source.ocIsKindOf(emendo::GameDesignElement) and target.ocIsKindOf
  (emendo::Belohnungsregel)
2
3 aql:target.ocIsKindOf(emendo::GameDesignElement) and source.ocIsKindOf
  (emendo::Belohnungsregel)

```

Diese Constraint-Ausdrücke bestehen dabei aus zwei Teilen, die jeweils beide vor dem Erstellen einer Zuordnungsverbindung zwischen den beiden genannten Elementtypen durch Sirius überprüft werden. Der erste Teil des Ausdrucks deckt den Fall ab, dass zuerst ein *GameDesignElement* (source) durch einen Nutzer ausgewählt wurde und das Ziel eine *Belohnungsregel* (target) darstellt, wohingegen der zweite AQL-Ausdruck den umgekehrten Fall abbildet. Da sich die AQL-Überprüfung direkt auf die Ecore Definition bezieht, und diese das objektorientierte Konzept der Vererbung unterstützt, kann in diesem Fall die Überprüfung auf Typebene der *GameDesignElemente* stattfinden und impliziert damit ebenso alle Kindklassen dieses Typs, wie beispielsweise *Badge* oder *Punktesystem*. Handelt es sich um den Versuch eine Zuordnungsverbindung zwischen zwei Notationselementen zu erstellen, die per Definition nicht miteinander verbunden werden dürfen, also die Überprüfung durch die Constraint-Ausdrücke ein negatives Ergebnis zurückliefert, wird dies dem Benutzer entsprechend kenntlich gemacht (siehe Abbildung 55, a).

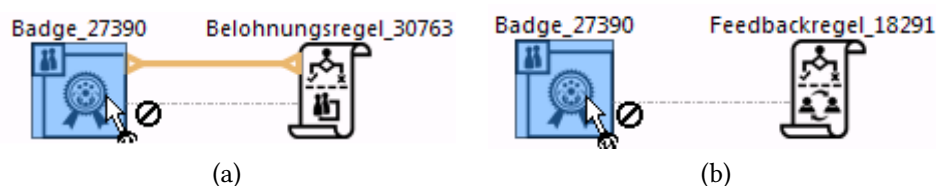


Abbildung 55: Visuelle Rückmeldung der Überprüfungen bei der Erstellung von Zuordnungsverbindungen

Darüber hinaus muss bei der Erstellung einer Zuordnungsverbindung als weitere Vorbedingung überprüft werden, ob bereits eine derartige Verbindung zwischen zwei Elementen besteht. Ist dies der Fall, darf keine weitere Zuordnungsverbindung gemäß der abstrakten Syntax erstellt werden und demnach der Designer dies auch nicht zulassen. Zur Umsetzung dieser Anforderung

wurde ein Java-Service (*istGueltigeZuordnungsverbindung(...)*) erstellt, der diese Überprüfung vornimmt abhängig von dem boolschen Ergebnis, das dieser Service zurückliefert, die Erstellung einer derartigen Verbindung zulässt oder nicht und dies dem Benutzer zurückmeldet (siehe Abbildung 55, b).

#### 6.4.2 Korrektheitsüberprüfungen bei Pflicht- und Optionalverbindungen

Obige Überprüfung, sowohl was das Quell- und Zielelement einer Pflicht- bzw. Optionalverbindung, als auch die Überprüfung, ob eine Verbindung des selben Typs zwischen zwei Elementen bereits besteht, muss vor der Erstellung dieser Verbindungstypen durchgeführt werden. Hierzu wurden jeweils Java-Services implementiert, die dies abbilden.

Da es sich bei diesen Verbindungstypen um gerichtete Verbindungen handelt, ist hier ein zusätzlicher Fall zu beachten, nämlich die Vermeidung von zyklischen Abhängigkeiten. Diese sind im Besonderen bei Pflichtverbindungen kritisch, da sie eine kausale Abhängigkeit zwischen zwei Lernelementen darstellen. Bei Optionalverbindungen dürfen jedoch ebenso keine Zyklen vorkommen, da dies in der späteren Anwendung des Modells in einem LMS zu Problemen führen kann, die das Vorschlagswesen von Lernbausteinen für Lernende betreffen.

Al-Mutawa et al. unterscheiden in symmetrische und asymmetrische Typen von zyklischen Abhängigkeiten (Al-Mutawa et al., 2014, S. 51). Durch die Unterbindung von einfachen Zyklen zwischen zwei Elementen (siehe Abbildung 56, a) sowie von kreisförmigen Zyklen (siehe Abbildung 56, b), ist es möglich, alle weiteren (Clique, Chain und Star (Al-Mutawa et al., 2014, S. 51)) symmetrischen Zyklen nach Al-Mutawa et al. (2014) auszuschließen.

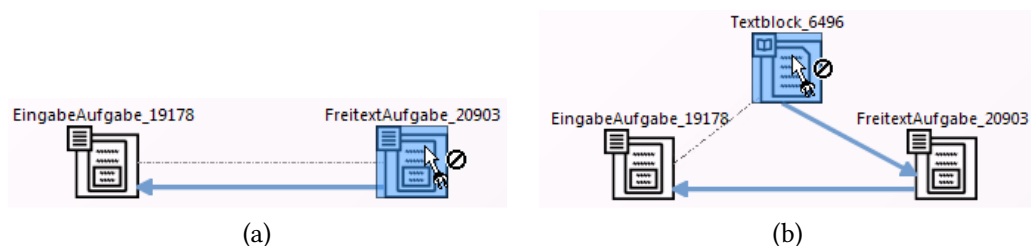


Abbildung 56: Vermeidung von symmetrischen zyklischen Abhängigkeiten bei Pflichtverbindungen

Die dafür notwendigen Überprüfungen wurden mit einem Java-Service entsprechend umgesetzt.

Zudem wurde eine Übersicht über alle möglichen Verbindungen zwischen den in der Emendo DSML vorhandenen Notationselementen erstellt. Diese Übersicht befindet sich im Anhang A.4 der vorliegenden Arbeit.

#### 6.4.3 Syntaktische Validierung von Bezeichnern

Neben den automatischen Überprüfungen, die OCL erlaubt, ob beispielsweise nicht leere Attributwerte bei Objekten auch wirklich nicht leer sind, gibt es darüber hinaus die Möglichkeit einzelne Attributwerte zu validieren. Beispielsweise ist dies sinnvoll, wenn ein Bezeichner eines Notationselements immer dasselbe Format besitzen soll, bestehend aus dem Namen des Notationselements, einem tiefgestellten Bindestrich, gefolgt von einer für ein Modell eindeutig generierten Objekt-ID (siehe beispielsweise Abbildung 56, *a*, EingabeAufgabe\_19178). In der Emendo DSML ist dies derartig definiert, da beim späteren Generieren von Code Sprachkonventionen eingehalten werden müssen, die beispielsweise fordern, dass in Bezeichnern von Variablen keine Leerzeichen auftauchen oder bestimmte Zeichen darin nicht vorkommen dürfen. Um dies sicherzustellen, wurde ebenfalls ein entsprechender Java-Service implementiert, der nicht nur die Überprüfung bereits bestehender Bezeichner übernimmt (Bezeichner können manuell durch Benutzer editiert werden), sondern initial beim Erstellen eines Notationselements den Wert dieses Features automatisch setzt.

#### 6.4.4 Automatisches Setzen von Attributwerten

Eine weitere Möglichkeit zur Validität von Modellen beizutragen, ist das automatische Setzen (bzw. das automatische Leeren) von Attributwerten, abhängig von Benutzeraktionen. Beispielhaft kann hier eine Benutzeraktion angeführt werden, bei der Benutzer einen *Hinweis* einer *EingabeAufgabe* mit einer Zuordnungsverbindung zuordnen (siehe Abbildung 57).

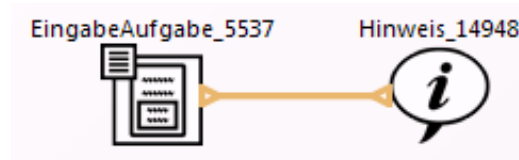


Abbildung 57: Zuordnung von Hinweis zu Eingabe-Aufgabe

Was im Emendo Designer intern passiert, ist jedoch mehr als das Zeichnen und Positionieren der Zuordnungsverbindung zwischen den beiden Notati-

onselementen. Vielmehr werden Attributwerte automatisch gesetzt, gemäß der in Ecore festgelegten Attribute (Feature) und deren für jedes Attribute definierten Einschränkungen. Wird im oben beispielhaft beschriebenen Fall eine Zuordnungsverbindung erstellt, werden zunächst die Attribute *quellModellelement* und *zielModellelement* der Verbindung mit Werten belegt. Diese ergeben sich durch die Reihenfolge, in der die Zuordnungsverbindung gezeichnet wurde, also das *quellModellelement* den Wert der Sirius-Variable *source* annimmt und das *zielModellelement* den Wert der Variable *target*. In einem nächsten Schritt erfolgt eine Überprüfung, ob es sich bei der *source* um ein Objekt des Typs *Aufgabe* handelt und beim *target* um ein Objekt des Typs *Hinweis*. Ist dies der Fall, erlaubt Sirius die Definition eines Kontextwechsels<sup>5</sup>, womit in den Kontext von Variablen gesprungen werden kann (Obeo GmbH, 2017b). In diesem Fall würde ein Kontextwechsel in den Kontext der *source*-Variable (also aus Sicht der *Aufgabe*) dazu führen, dass der Wert der Aufgaben-Variable *loesungstipp* auf den Wert der *target*-Variable (das Hinweis-Objekt) gesetzt würde. Sollte die Belegung von *source* und *target* unmittelbar nach der Erstellung der Verbindung und vor dem Setzen der jeweiligen Attribute vice versa erfolgt sein, werden die Attributwerte entsprechend auch vice versa gesetzt.

In ähnlicher Weise verhält es sich bei gerichteten Verbindungstypen, mit dem Unterschied, dass nur das *quellModellelement* und das *zielModellelement* gesetzt werden müssen, da es keinen bidirektionalen Bezug zwischen zwei Notationselementen gibt, die mit einer Optional- oder Pflichtverbindung verbunden werden. Anders ausgedrückt, kennt ein Element A nur seine(n) Nachfolger B und B wiederum nur seine(n) Nachfolger C.

Nicht nur das Setzen von Attributwerten aufgrund von Verbindungen findet im Emendo Designer Berücksichtigung, sondern auch das Löschen der gesetzten Werte. Demnach existieren die beschriebenen Operationen auch für das Löschen von Verbindungen und das damit verbundene Löschen von gesetzten Attributwerten bei den verbundenen Objekten. Darüber hinaus ist zu beachten, dass eine Löschung der Zuordnungsverbindung in dem oben beschriebenen Fall zwar zu einem Leeren der entsprechenden Attributwerte führt, eine Löschung des Hinweises jedoch impliziert, dass die Zuordnungsverbindung ebenfalls gelöscht werden müsste, da Zuordnungsverbindungen per Definition immer ein Quell- und Zielelement besitzen müssen. Fehlt eines dieser Element durch Löschung, wird die Zuordnungsverbindung dadurch

<sup>5</sup> Ein Kontextwechsel stellt in Sirius die Möglichkeit dar, Code aus Sicht eines bestimmten Programmbestandteils auszuführen, auf dem der Steuerungsfokus liegt, also beispielsweise aus Sicht einer Variable.

ebenfalls ungültig und muss deshalb entfernt werden. Hierbei wurden ebenfalls Java-Extensions eingesetzt, die ein korrektes Löschen und Leeren der entsprechenden Attributwerte sicherstellen.

#### 6.4.5 Kontextabhängige Definition von Objektzuständen über Eingabedialoge

Für Notationselemente besteht aus Sicht der Benutzer die Möglichkeit, per Doppelklick auf ein Element, dieses mit Inhalten zu detaillieren und damit Attributwerte zu setzen. Dies geschieht über JFace-Dialoge, deren Start über ein Tool in Sirius definiert wurde und mit Hilfe einer Java-Extension, abhängig vom jeweiligen Typ des Notationselements, der entsprechende JFace-Dialog gestartet wird. Beispielfhaft sind in Abbildung 58 zwei Dialogtypen dargestellt: ein Dialog, der eine *EingabeAufgabe* detailliert (Abbildung 58, a) sowie der Dialog, mit dem eine Detaillierung einer Belohnungsregel vorgenommen werden kann (Abbildung 58, b).

Für alle weiteren Notationselemente der Emendo DSML existieren ebenfalls derartige teils ineinander geschachtelte Dialoge (beispielsweise ist eine rekursive Schachtelung der Dialoge aufgrund der rekursiven Struktur von Bedingungen von Regeln notwendig), ausgenommen für Elemente der Kategorie Verbindungen und Strukturen.

Für den Fall, dass an das Notationselement der Belohnungsregel aus Abbildung 58 b, vor dem Starten des Dialogs ein Punktesystem angehängt wurde, werden daraufhin die für den Benutzer auswählbaren Vorschläge des Regeldialogs auf den Kontext des Punktesystems beschränkt. Somit bewirkt die Zuordnungsverbindung zwischen den beiden Notationselementen eine entsprechende Einschränkung der möglichen Bedingungen und Konsequenzen der Regel. Selbiges gilt für eine Zuordnungsverbindung zwischen einem Lernelement, wie die Eingabe-Aufgabe eines darstellt und der Belohnungsregel, in Bezug auf die Benutzeraktion. Da Regeln durch Aktionen auf Lernelementen getriggert werden, wird die mögliche Auswahl an Benutzeraktionen durch eine derartige Verbindung im Dialog der Belohnungsregel entsprechend auf den Kontext der Eingabe-Aufgabe eingeschränkt. Damit ist es nicht möglich für Benutzer Wertbelegungen von Attributen von Objekten in Dialogen auszuwählen, die semantisch nicht für die aktuell in Beziehung stehenden Objekte valide sind. Dies trägt maßgeblich zu einer gesteigerten Validität von Modellen bei.

Aus technischer Sicht erfolgte diese kontextbasierte Filterung als Teil der jeweiligen Dialogklassen. Die im Metamodell der Emendo DSML eingefügten Zwischenebenen (z. B. die Kapselung von den konkreten Klassen *AufgabeWie-*

(a)

(b)

Abbildung 58: Beispiele für JFace-Dialoge zur Detaillierung von Objektinformationen

*derholbar* und *AufgabeKorrekt* unter der abstrakten Elternklasse *AufgabenBoolscheBedingungsvariable*) erleichtern dabei die korrekte Filterung der zur Verfügung stehenden Auswahlmöglichkeiten für den Benutzer, aufgrund von Typüberprüfungen im Code (ähnlich wie im Filter Pattern<sup>6</sup>, nur mit einem

<sup>6</sup> Siehe z. B. Singh (2016).

einziges Kriterium, das sich auf den Typ eines verbundenen Objekts bezieht). Wenn sich Änderungen im Modell ergeben, die sich entsprechend auf die auswählbaren Einträge der Dialoge auswirken, so werden die für die Auswahl nötigen Überprüfungen bei jedem Dialogstart durchgeführt, was die Korrektheit der Objektzustände weiterhin unterstützt. Das Abrufen aktuell gesetzter Attributwerte sowie das Setzen neuer Attributwerte erfolgt über die durch das EMF-Framework bereitgestellten und aus Ecore generierten Interface- bzw. Implementierungsklassen und den darin verfügbaren Standardzugriffsmethoden.

#### 6.4.6 Modellweite Synchronisation von Punkte- und Levelsystem

Eine weitere Maßnahme zur Sicherstellung der Validität von Modellen stellt die Modell-weite Synchronisation von Daten des Punkte- bzw. Levelsystems dar. In Anlehnung an die Anforderungen P\_A1 und L\_A4 aus Tabelle 15 in Kapitel 4, darf es pro Instanz einer Lerneinheit nur ein einheitliches Punkte- bzw. Levelsystem geben. Werden beispielsweise zwei Punktesysteme mit dem Notationselement modelliert, erstellt Sirius bzw. EMF im Hintergrund zwei verschiedene Objekte, die es zulassen, ihre jeweiligen Attribute unterschiedlich mit Werten zu belegen. Dies könnte dazu führen, mehrere Punkttypen innerhalb einer Lerneinheit im LMS an Lernende vergeben werden, oder im Falle von Levelsystemen, unterschiedliche parallel progressiv steigende Level existieren, die keinen Zusammenhang zueinander aufweisen. Um die beiden genannten Fälle zu unterbinden, wurde sich des durch EMF bereitgestellte Beobachter-Mechanismusses für Änderungen an Objektzuständen bedient. Hierfür wurden zwei *NotificationFilter* implementiert, die sich als konkrete Beobachter an dem *RessourceSet* (so bezeichnet EMF das Datenmodell, was durch einen Benutzer erstellt wird) registriert haben, um Objektzustandsänderungen von Punkte- oder Levelsystemobjekten separat zu behandeln. Bei jeder Änderung an einem modellierten Punkte- oder Levelsystem wird der konkrete *NotificationFilter* getriggert und leitet die Ereignisverarbeitung an einen *ModelChangeTrigger* weiter. Letzterer identifiziert die Art der Änderung an einem Objekt und sorgt dafür, dass auf allen anderen im Modell verfügbaren Objekten des selben Typs, die selbe Änderung vollzogen wird. Wird also beispielsweise durch einen *ModelChangeTrigger* festgestellt, dass eine *punktesystemBezeichnung* eines *Punktesystem*-Objekts gesetzt wurde, wird die entsprechende Änderung in Art und Wert identifiziert und ein transaktionsbasiertes EMF-Kommando gemäß dem Command Pattern (Gamma et al., 1995, S. 263ff.) erstellt, dass das Setzen dieses Werts bei allen verfügbaren Objek-



ten des *Punktesystem*-Typs durchführt. Die entsprechende Java-Klasse *PunktesystemSynchronizationChangeTrigger* für die Behandlung von Punktesystem-Objekten ist in Code-Ausschnitt 10 beispielhaft aufgeführt. Das Verhalten für Levelsystem-Objekte ist auf Code-Ebene vergleichbar und unterscheidet sich lediglich im Typ des Filter-Objekts.

Code 10: Konkreter ModelChangeTrigger zur Synchronisation von Punktesystem-Objekten

```

1 package org.eclipse.emendo.design.triggers;
2
3 // imports, die aus Gruenden der Uebersichtlichkeit entfernt wurden
4
5 public class PunktesystemSynchronizationChangeTrigger implements
6     ModelChangeTrigger {
7
8     @Override
9     public int priority() {
10         return 0;
11     }
12
13     @Override
14     public Option<Command> localChangesAboutToCommit(Collection<
15         Notification> notifications) {
16         CompoundCommand compoundCommand = new CompoundCommand();
17         for (Notification notification : notifications) {
18             int eventType = notification.getEventType();
19             Object changedFeature = notification.getFeature();
20
21             if (eventType == Notification.SET
22                 && changedFeature instanceof EAttribute
23                 && changedFeature != EmendoPackage.Literals.MODELLEMENT__ID)
24             {
25                 EAttribute changedAttribute = (EAttribute)changedFeature;
26
27                 Object notifier = notification.getNotifier();
28                 if (notifier instanceof Punktesystem) {
29                     Punktesystem punktesystem = (Punktesystem) notifier;
30                     Session session = SessionManager.INSTANCE.getSession(
31                         punktesystem);
32                     TransactionalEditingDomain editingDomain = session.
33                         getTransactionalEditingDomain();
34
35                     EObject eContainer = punktesystem.eContainer();
36
37                     if (eContainer instanceof EmendoModell) {
38                         EList<Modellement> modellemente = ((EmendoModell)
39                             eContainer).getModellemente();
40
41                         for (Modellement modellement : modellemente) {
42                             if (modellement instanceof Punktesystem) {

```

```

38         compoundCommand.append(SetCommand.create(editingDomain,
39             modelElement, changedAttribute, notification.
40             getNewValue()));
41     }
42 }
43 }
44 }
45
46 if (compoundCommand.isEmpty()) {
47     return Options.newNone();
48 } else {
49     return Options.newSome(compoundCommand);
50 }
51
52 }
53
54 }

```

## 6.5 EMENDO GENERATOR

Im Folgenden soll auf den Emendo Generator eingegangen werden. Dieser wird als Teil des Emendo Designers beschrieben, da seine Funktionalität über den Designer verwendet werden kann. Dies bedeutet, dass das Generieren eines Modells, das mit der Emendo DSML erstellt wurde, direkt aus dem Emendo Designer heraus angestoßen werden kann.

### 6.5.1 *Integration in den Emendo Designer*

Um ein mit der Emendo DSML erstelltes Modell in Quellcode übersetzen zu lassen, wird ein für die Emendo DSML erstellter und Template-basierter Acceleo<sup>7</sup> Generator verwendet. Das Anstoßen des Validierens und Generierens eines Modells wurde über Erweiterungen des Menüs und der Toolbar umgesetzt. Dafür wurden Eclipse-Bindings definiert, die die entsprechenden Menüeinträge und Toolbar-Icons mit Handlern verbinden. Handler sind im Kontext von Eclipse konkrete Beobachter für UI-Elemente, bekannt aus dem Beobachtermuster (Gamma et al., 1995, S. 326ff.). Wird der Generierungshandler aufgerufen, startet dieser zunächst eine EMF-Validierung des Modells, um zu gewährleisten, dass nur syntaktisch korrekte Modelle generiert werden können und damit Fehler im Vorfeld erkannt werden. Ist die Validierung des Modells erfolgreich und im Ergebnis fehlerfrei, hat der Benutzer die Möglichkeit einen

<sup>7</sup> Version 3.6.6.

Zielpfad für den Output des Generators anzugeben. Nach der Auswahl dieses Pfades beginnt die eigentliche Generierung, wobei die entsprechende Funktionalität von Acceleo wiederum über zwei Plug-Ins in den Emendo Designer integriert ist und dessen Funktionalität damit erweitern.

### 6.5.2 Struktur des Generators

Der bereits in den Grundlagen zu Generatoren diskutierte Template-Mechanismus spiegelt sich in zwei voneinander verschiedenen Modulen wieder: Ein Hauptmodul *emendogenerate*, welches die Java-Klasse *EmendoModell.java* generiert (und damit die zentrale Code-Repräsentation eines Emendo Modells darstellt) und sich für seine Ausführung der darin definierten Funktionalität des Moduls *emendoservices* bedient. Letzteres enthält Weiterleitungen von Funktionsaufrufen, die aus dem Hauptmodul des Acceleo Generators heraus in Java-Klassen zur Bearbeitung weitergegeben (invoked) werden. Dies ist in solchen Fällen hilfreich, wenn für OCL zu komplexe Überprüfungen benötigt werden und sich des Sprachumfangs einer Programmierhochsprache, wie Java bedient werden muss.

Der grundlegende Algorithmus, zum Traversieren und Generieren von Emendo-Modellen, ist *Preorder* (Sedgewick u. Wayne, 2014, S. 877ff.). Preorder bedeutet, dass die einzelnen Templates in den Modulen einem Schema folgen, wonach eine zu erstellende Java-Klasse *EmendoModell* bezogen auf eine Modellinstanz immer zunächst bei der größten Einheit (abgesehen von Klassenrumpf und Variablendeklarationen sowie Konstruktor) mit dem Generieren begonnen werden muss, bevor die nächst kleinere Einheit generiert wird und zu der bestehenden größeren Einheit als Teil hinzugefügt wird. Dieses Paradigma ist einfach an einem Beispiel zu veranschaulichen: Soll eine Belohnungsregel erstellt werden, die aus einer Benutzeraktion, einer Bedingung und einer Belohnungskonsequenz besteht, so muss zunächst die Regel (Wurzel eines Teilbaums) im Code allgemein deklariert und initialisiert werden. Anschließend müssen alle drei Teile (Blätter oder weitere Wurzeln eines Teilbaums) einer Regel jeweils vollständig aus einem entsprechenden Modell extrahiert und generiert werden, um sie jeweils anschließend der bereits generierten Regel hinzuzufügen. Der Code-Auszug 11 zeigt dieses Verhalten, abgebildet in einem Acceleo-Template, als Teil des Hauptmoduls des Emendo Generators. Demnach ist es nicht verwunderlich, dass das Traversieren und Generieren eines Emendo Modells bei den Lernelementen vom Typ Gruppe

beginnt, da jedes Modell genau eine Gruppe als Wurzel enthält, in der alle weiteren Lernelemente als Bestandteile vorkommen.

Code 11: Template Auszug aus dem Hauptmodul von des Emendo Generators zur Generierung einer Belohnungsregel

```

1 [template public createRegelFuerLernelement (r : Regel, lernelement :
   Lernelement) ]
2 // Belohnungsregeln
3 [if (r.ocIsTypeOf(Belohnungsregel))]
4 [let br : Belohnungsregel = r]
5   [createBelohnungskonsequenzen(br)/]
6 [for (ausdruck:ecore::EString | br.regelAusdruecke)]
7 IAusdruck bedingungsString_[br.sichtbarerBezeichner/][br.
   regelAusdruecke.indexOf(ausdruck)/] = AusdrucksParser.fromString("[
   ausdruck/"]");
8 [/for]
9 Regel regel_[r.id/] = new Regel.Builder()
10   .mitBenutzeraktion(Benutzeraktion.[r.benutzeraktion.eClass().name/])
11   [for (ausdruck:ecore::EString | br.regelAusdruecke)]
12     .mitAusdruck(bedingungsString_[br.sichtbarerBezeichner/][br.
       regelAusdruecke.indexOf(ausdruck)/])
13   [/for]
14   [for (bk : Belohnungskonsequenz | br.belohnungskonsequenzen)]
15     .mitKonsequenz(konsequenz_[br.sichtbarerBezeichner/][bk.befehlsName
       .toUpperFirst()/])
16   [/for]
17   .build();
18 [/let]
19 [/if]
20 regel_[r.id/].setRegelId([r.id/]);
21 [lernelement.sichtbarerBezeichner/].addRegel(regel_[r.id/]);
22 [/template]

```

In besagter Gruppe wird anschließend der erste Lernbaustein identifiziert, der keine eingehenden Pflicht- und Optionalverbindungen besitzt, kein Unlock-Element ist sowie nicht mit einer Zuordnungsverbindung einer Aufgabe zugeordnet ist. Ist dieser gefunden, wird der Lernbaustein und seine Bestandteile sowie Zuordnungen erzeugt und der Modell-Baum weiter in Preorder-Reihenfolge durchlaufen.

### 6.5.3 Anbindung an Domain Framework

In dem Code Auszug 11 ist zudem zu erkennen, dass sich der Generator der Funktionalität des Domain Frameworks bedient, wie beispielsweise dem Aufruf des *Regelbuilders*, der die Funktionalität bereitstellt, eine Regel anhand ihrer Bestandteile zu erstellen, oder des *AusdrucksParsers*, der in der Lage ist, die einzelnen Bedingungszeichenketten zu parsen und entsprechend in Klas-

sen des Domain-Frameworks zu überführen.

Eine Generierung für unterschiedliche Domain Frameworks, also verschiedene Zielplattformen, stellt mit dem Emendo Generator und der Emendo DSML keine Probleme dar. Dies liegt vor allem an einer konsequenten Trennung von Daten und deren Funktionalität, die im Metamodell der DSML definiert und schließlich im Generator umgesetzt wird. Domänenobjekte, wie Lernbausteine oder Regeln, werden mit samt ihren Daten und ihrem Verhalten, welche für eine vollständige und korrekte Umsetzung in eine spätere Zielplattform notwendig sind, lediglich durch Emendo Modelle beschrieben und mit einem *M2T* Generator in Quellcode übersetzt. Die eigentliche Nutzung dieser Informationen obliegt der Zielplattform, also in diesem Fall dem Emendo LMS. Durch diese strikte Trennung von Beschreibung und Ausführung ist die Wahl der Zielplattform vergleichsweise flexibel, sofern diese mit dem generierten Modell und den darin enthaltenen Informationen umzugehen weiß. Damit liegt die Vermutung nahe, dass eine Wiederverwendung der Konzepte aus dem Metamodell der DSML in dem Domain Framework durchaus sinnvoll ist, um unter Umständen komplexe Abbildungen von Typen von Definitionen im Metamodell hin zu Definitionen im Domain Framework zu vermeiden. Folglich muss sich nicht die gesamte Struktur des Metamodells im Domain Framework wiederfinden, jedoch für Klassen des Domain Frameworks gelten, dass sie mindestens die Daten und Funktionalität bereitstellen, die bereits im Metamodell definiert wurde. Wie dies im Emendo LMS ausgestaltet ist, soll im folgenden Kapitel thematisiert werden. Zuvor soll sich jedoch verwandten Arbeiten zu der vorliegenden gewidmet werden, um Emendo einerseits abzugrenzen und andererseits einen Überblick über aktuelle Forschungsarbeit in diesem Gebiet zu geben.

## 6.6 VERWANDTE ARBEITEN

Neben der vorliegenden Arbeit existieren weitere relevante Arbeiten, die im Folgenden Erwähnung finden sollen. Grundsätzlich ist zu sagen, dass kein direkt vergleichbarer derartiger Ansatz auf Basis von DSM und im Kontext von hochschulischer sowie spielifizierter Lehre zum Zeitpunkt der Anfertigung dieser Arbeit dokumentiert ist. Eine Möglichkeit zur Unterteilung der existierenden und verwandten Arbeiten ist die Klassifizierung in textuelle und grafische domänenspezifische (Modellierungs-) Sprachen. Dieser Unterteilung soll sich im Folgenden bedient werden.

### 6.6.1 *Game Description Language und ihre Weiterentwicklung*

Eine textuelle domänenspezifische Sprache wurde von Thielscher (2010) auf Basis der von Love et al. (2008) entwickelten Game Description Language (GDL) erstellt<sup>8</sup>. Bei der GDL handelt es sich um eine formale und generalisierte Sprache, die verwendet wird „[...] to represent the rules of  $n$ -player games ( $n \geq 1$ ) in such a way that they can be automatically processed by a general game player“ (Thielscher, 2010, S. 994). Ein allgemeiner Spieler wird durch eine derartige formale Sprache in die Lage versetzt, die Regeln eines unbekannten Spiels zu durchdenken, um Spiel-spezifisches Wissen daraus zu extrahieren und aufgrund dessen Spielentscheidungen zu treffen (Thielscher, 2010, S. 994). Dabei wird bei der GDL unterstellt, dass das Spiel zwar unbekannt sein kann, jedoch dies deterministisch sein muss sowie für Spieler jederzeit über vollständige Informationen bzgl. des Spielstatus verfügt (z. B. bei Schach) (Thielscher, 2010, S. 994). Thielscher entwickelt diese Restriktion in der GDL zu einer GDL-II weiter, indem er diese Restriktion aufhebt und die GDL-II Anwendung in Spielen finden kann, bei der der Spielstatus nicht allen Spielern bekannt sein muss (z. B. bei Poker) (Thielscher, 2010). Bei der GDL-II handelt es sich um eine wohldefinierte textuelle domänenspezifische Sprache, welche die GDL syntaktisch und semantisch erweitert.

Die GDL-II lässt sich beispielhaft an einem Schachspiel<sup>9</sup> erklären. Bei einem klassischen Schachspiel, ist der jeweilig gegnerische Zug für einen Spieler erkennbar und würde in GDL-II formal beschrieben mit dem Code Auszug 12.

Code 12: Formale Beschreibung von Schach mit GDL-II (vollständige Information)

```

1 sees(white,M)  $\Leftarrow$  does(black,M)
2 sees(black,M)  $\Leftarrow$  does(white,M)

```

Dabei ist  $R$  ein Spieler, der einen Zug  $M$  durchführt ( $does(R,M)$ ), bzw. einen Zug  $P$  wahrnimmt ( $sees(R,P)$ ). Weitere Schlüsselwörter der GDL bzw. GDL-II sind zu finden in Love et al. (2008) bzw. Thielscher (2010).

In einer Variante, in der für den gegnerischen Spieler der Zug des Gegners nicht ersichtlich ist (unvollständige Information über Spielstatus), würde ein Schiedsrichter benötigt, der entscheidet, ob es sich bei einem Zug um einen korrekten Zug im Sinne der Spielregeln handelt. Der Code Auszug 13 zeigt beispielhaft die Umsetzung in GDL-II.

<sup>8</sup> Präziser gesagt, handelt es sich bei Thielschers Arbeit um eine Weiterentwicklung der von Love et al. (2008) erstellten GDL.

<sup>9</sup> Beispiel entnommen aus (Thielscher, 2010, S. 995f.).

Code 13: Formale Beschreibung von Schach mit GDL-II (unvollständige Information)

```

1 sees(R,badMoveTryAgain)  $\Leftarrow$  does(R,M)  $\wedge$   $\neg$  validMove(M)
2 sees(black,yourMoveNow)  $\Leftarrow$  does(white,M)  $\wedge$  validMove(M)
3 sees(white,yourMoveNow)  $\Leftarrow$  does(black,M)  $\wedge$  validMove(M)

```

Obwohl es sich bei der GDL um eine ausdrucksstarke domänenspezifische Sprache handelt, ist sie nicht dafür geeignet neue Spiele zu entwickeln, sondern vielmehr bestehende zu beschreiben. Die textuelle Notation von GDL erinnert an eine aussagenlogische Anwendung und ist damit für nicht technisch-versierte Benutzer nicht einfach zu erlernen. Außerdem ist die GDL nicht dafür vorgesehen, Modellinstanzen automatisch auszuführen oder weiter zu verarbeiten (vielmehr zu interpretieren), da sie nicht auf einem wohldefinierten Metamodell basiert, was sie von der vorliegenden Arbeit unterscheidet. Zudem fokussiert sie Spiele im Allgemeinen und nicht Spielifizierung durch Gamification.

### 6.6.2 Card Game Description Language

Die Card Game Description Language (CGDL) von Font et al. (2013) stellt eine formale, textuelle und ausführbare domänenspezifische Sprache zur Definition von Kartenspielen, wie Poker Texas hold 'em oder UNO dar. Die CGDL basiert auf einer kontextfreien Grammatik, die mit Hilfe der Backus-Naur-Form (BNF) definiert wurde (Font et al., 2013, S. 258). Die textuelle Sprache unterscheidet in vier Regeltypen (operational, play-only-once, mandatory und Computer commands (Font et al., 2013, S. 257f.)), jeweils bestehend aus Bedingungen und Konsequenzen, welche als Spielmechaniken während eines Kartenspiels mit mehr als einem Spieler zum Einsatz kommen. Ein Kartenspiel ist zudem definiert durch „[...] a set of stages, a ranking of card combinations, and a set of winning conditions“ (Font et al., 2013, S. 257). Die Syntax der CGDL ist beispielhaft für das Kartenspiel UNO in Abbildung 59 dargestellt.

Das Kartenspiel UNO wird in der CGDL in zwei Stages unterschieden (siehe Abbildung 59). In der ersten Stage (*Stage 0*), gibt der Computer (*com*) an alle Spieler 7 Karten aus und legt in einem zweiten Schritt vom Deck eine Karte an Position *TO*. In der zweiten Stage (*Stage 1*) werden Karten auf Position *TO* gezeigt (*show*) und von Spielern ausgespielt (*playit*), die entweder die gleiche Farbe (*same suit*) oder die gleiche Zahl (*same number*) als jeweilige Bedingung besitzen. Als nächstes wird eine Karte vom Deck durch Spieler gezogen (*draw*) und der nächste Spieler ist an der Reihe (*next*). Zuletzt wird mit einer

<i><b>STAGES</b></i>	
NAME	RULES
STAGE 0	com - deal, ALL PLAYERS, 7 com - deal, T0, 1
STAGE 1	show, same suit, T0 - playit show, same number, T0 - playit draw - next mandatory_have, λ - win

Abbildung 59: Auszug aus der Syntax der CGDL bei der Definition von UNO (Font et al., 2013, S. 261)

*mandatory*-Regel, ohne Bedingung festgelegt, wer das Spiel gewinnt.

Die CGDL ist eine auf Kartenspiele fokussierte domänenspezifische Sprache, die nicht für den Einsatz im Kontext von Gamification oder Lehre erstellt wurde. Es handelt sich zudem um eine beschreibende Sprache, die weniger für Neuentwicklungen intendiert ist, sondern vielmehr bereits bestehende Kartenspiele auf Basis von grundlegenden Axiomen deklarativ und auf Basis eines Metamodells beschreibt (Font et al., 2013, S. 257). Diese Beschreibungen können für Simulationen verwendet werden, was die Sprache ausführbar macht, auch wenn die Definition der dafür notwendigen textuellen Modelle durch deren formale Notation für nicht-technisch affine Benutzer erschwert ist (Font et al., 2013, S. 361f.).

### 6.6.3 Gamification Modeling Language

Ein weiterer textueller Repräsentant, der im Folgenden Erwähnung finden soll, ist die Gamification Modeling Language (GaML) (Herzig et al., 2013). Diese formale textuelle Modellierungssprache basiert auf einer abstrakten Syntax, deren Grammatik mit Hilfe der Erweiterten Backus-Naur-Form (EBNF) definiert wurde und durch statische semantische Ausdrücke während des Erstellens von textuellen Modellen anhand des Syntaxbaumes validiert wird (Herzig et al., 2013, S. 497). Mit der GaML können allgemeine Gamification-Konzepte mit xText definiert werden, welche eine Ausführung der damit erstellten textuellen Modelle erlauben (Herzig et al., 2013). Die Autoren geben an, die Sprache für den Geschäftseinsatz nutzen zu wollen, konkret für die Einbettung



spielifizierter Einheiten (als *SAP HANA Software-as-a-Service*) in die *SAP Gamification Plattform*<sup>10</sup> (Herzig et al., 2013, S. 499).

GaML unterscheidet in atomare (z. B. Punkt, Mission, Feedback, Benachrichtigung) und aggregierte (z. B. Kontext, Avatar, Kommunikationssystem) Game Mechanics (Herzig et al., 2013, S. 495). Diese Mechanics werden über wenn-dann-Regeln textuell miteinander verknüpft. Der Code Auszug 14 zeigt eine beispielhafte Definition eines Modells<sup>11</sup>, dass einen *Carpool* (*SAP TWOGO*<sup>12</sup>) abbildet, die Benutzeraktion *ride* deklariert und mit Properties definiert, ein internes Event *socializerEvent* mit zwei String-Properties sowie eine atomare Mechanic vom Typ *point* mit dem Namen *Socializer Point* und der Abkürzung *SP*. Außerdem wird eine aggregierte Mechanic vom Typ *mission* deklariert und darin zwei Regeln definiert.

Code 14: Beispielhaftes textuelles GaML-Modell eines Carpools

```

1 concept CarPooling {
2   useraction ride {
3     properties {
4       carbonoxid:Decimal,
5       kilometers:Decimal,
6       driver:Boolean,
7       rideId:Number
8     }
9   }
10
11   internalevent socializerEvent {
12     properties {pid:Text, friend:Text}
13   }
14
15   point Socializer {
16     name="Socializer Point", abbreviation="SP"
17   }
18
19   mission GetSocial {
20     when {
21       evt1:useraction(type=ride)
22       evt2:useraction(type=ride, rideId=evt1.rideId, player<>evt1.player
23         )
24       internalevent(type=socializerEvent, pid=evt1.player, friend=evt2.
25         player)
26     } then {
27       create event socializerEvent(pid=evt1.player, friend=evt2.player)
28     }
29     when {
30       internalevent(type=socializerEvent)
31     } then {

```

10 Siehe hierzu <https://cloudplatform.sap.com/capabilities/collaboration/gamification.html>.

11 Auszugweise entnommen aus (Herzig et al., 2013, S. 498).

12 Siehe hierzu: <https://www.twogo.com/>.

```

30     give Socializer points: 1
31   }
32 }
33 }

```

Der erste when-Clause besagt, dass wenn zwei Benutzeraktionen vom Typ *ride* auftauchen und diese den selben *ride* und den selben Spieler referenzieren, soll ein *socializerEvent* für die beiden Spieler gestartet werden. In dem zweiten when-Clause wird dieser Eventtyp abgefangen und jeder Spieler erhält ein *Socializer Point*. Bei dem vorgestellten Carpool-Beispiel geben die Autoren jedoch zu bedenken, dass die für GaML intendierte Zielgruppe nicht in der Lage ist, derartige Modelle zu erstellen (Herzig et al., 2013, S. 498).

Obwohl GaML eine generalisierte und durch ein Metamodell wohldefinierte Sprache für die Definition von Gamification-Einheiten darstellt, ist der Einsatz der Sprache nach eigenen Aussagen der Ersteller aufgrund der Komplexität limitiert (Herzig et al., 2013, S. 498). Auch ist eine Übertragbarkeit auf einen Lehr-Lernkontext nicht gegeben, da GaML eine generalisierte Sprache für die Definition von Gamification-Konzepten darstellt und die Bildungsdomäne unberücksichtigt lässt. Damit ist *GaML* zwar domänenspezifisch bezogen auf Gamification, jedoch nicht bezogen auf Lehr-Lernumgebungen und deren spezifischen Konzepte.

Zudem nutzt GaML eine textuelle Notation, was deren Elemente, visuell schwerer für den Nutzer erkennbar und differenzierbar macht als eine grafische Notation dies tun würde (Moody, 2009; Frank, 2013). Ebenso unterscheidet GaML in der Notation nicht zwischen den beiden Ebenen Struktur und Inhalt, sondern stellt beide gleichzeitig dar, was bei umfangreicheren Beispielen, als dem oben vorgestellten Minimalbeispiel eines Carpools, zu für Benutzer unübersichtlichen Modellen führen kann (Moody, 2009, S. 763, 769f.).

#### 6.6.4 *Machinations*

Als grafischer Repräsentant einer domänenspezifischen Sprache, ist das Framework *Machinations* von Dormans (2012) zu nennen. Es beschreibt Spiele formal durch ein Regel-basiertes und dynamisches System, bestehend aus Game Mechanics und deren Beziehungen miteinander (Dormans, 2012, S. 68). Zudem ist *Machinations* nicht nur für die theoretische Analyse bestehender Spiele geeignet, sondern ebenso für die Erstellung neuer Spiele (Dormans, 2012, S. 68). Die konkrete Syntax des Frameworks unterscheidet in die Notationselementtypen *Resource Connections* (geben an, wie Ressourcen zwischen



ßend in einer Zielumgebung zur Ausführung zu bringen (Dormans, 2012, S. 193). Da das Framework domänenspezifisch für Game Design ist, fokussiert Machinations die Domäne der Lehre nicht. Auch eine Eignung für diese Domäne in Verbindung mit Gamification ist als fraglich anzusehen, da Gamification ein Subset an Game Design-Elementen von Spielen darstellt, welche nicht alle mit Machinations abgebildet werden können (z. B. keine bedingte Vergabe von Badges geknüpft an Quests oder Teilen davon möglich). Zudem können, wie die Abbildung 60 zeigt, die Modelle, welche mit dem Framework erstellt werden, relativ komplex werden, was sich negativ auf die Effektivität bei deren Erstellung und Nutzung auswirken kann (Frank, 2013).

#### 6.6.5 ATTAC-L

Eine DSML und auf dem DSM-Ansatz basierte Arbeit stellen van Broeckhoven u. de Troyer bzw. de Troyer et al. mit *ATTAC-L* vor (van Broeckhoven u. de Troyer, 2013; de Troyer et al., 2017). Bei *ATTAC-L* handelt es sich um eine sogenannte Controlled Natural Language (CNL) (Wyner et al., 2010), was eine definierte Teilmenge einer natürlichen Sprache darstellt. Die DSML wurde für die Szenario-orientierte Definition von Serious Games mit dem Fokus auf Cybermobbing erstellt. Die Gesamtheit aus einzelnen Szenariodefinitionen bildet zusammen mit einem definierten Ablauf der einzelnen Szenarios und Entitäten (z. B. ein Opfer, ein Freund oder ein „Mobber“), die während dem Ablauf involviert sind, ein Story-Modell (van Broeckhoven u. de Troyer, 2013, S. 3). Zusätzlich sieht die Sprache elementare Aktionen vor, die als „Game Move“ bezeichnet werden (van Broeckhoven u. de Troyer, 2013, S. 3). Sie können von Spielern ausgeführt werden (z. B. jemanden begrüßen) oder eine Statusänderung einer Entität bedeuten (z. B. die Laune eines virtuellen Charakters ändert sich) (van Broeckhoven u. de Troyer, 2013, S. 3).

Die konkrete Syntax der DSML nutzt das Konzept von Bricks (wobei Bricks Game Moves repräsentieren) und ist laut den Machern angelehnt an die Notation von *Storybricks*<sup>14</sup>. Bricks sind wiederum unterteilt in verschiedene Kategorien (Objects, Object Classes, active/passive Objects, defined/anonymous Objects, negation of/passive Verbs und Adverbs (van Broeckhoven u. de Troyer, 2013, S. 4f.)), wobei sich jede Kategorie auf Wortkategorien aus der CNL bezieht und die Anordnungsstruktur einer EBNF-Grammatik folgt. Bricks werden in Kontrollstrukturen (Sequence, Choice, Order Independence und Con-

<sup>14</sup> Storybricks basiert ursprünglich auf dem vom MIT entwickelten Scratch (Resnick et al., 2009), jedoch wurde die Entwicklung 2014 gänzlich eingestellt.

currency (van Broeckhoven u. de Troyer, 2013, S. 5f.)) eingebettet bzw. damit verknüpft.

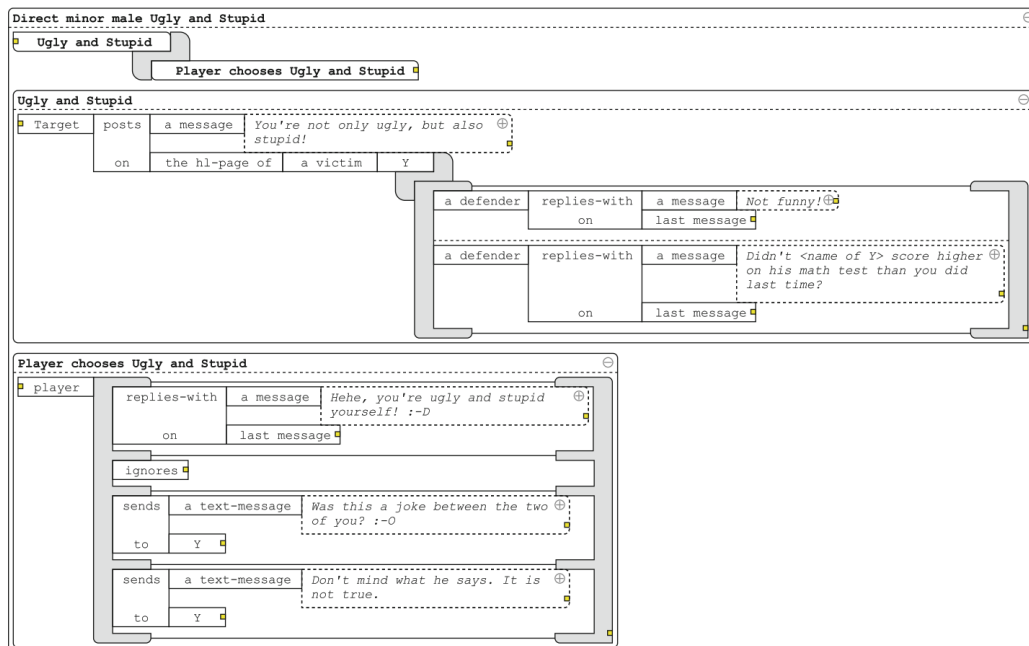


Abbildung 61: Beispiel für ein ATTAC-L Storyline Modell (de Troyer et al., 2017, S. 577)

Die Abbildung 61 zeigt ein Beispielmmodell einer Story, die in (Sub-) Szenarien (*Ugly and Stupid* und *Player chooses Ugly and Stupid*) unterteilt ist und in einem übergeordneten Szenario (*Direct minor male Ugly and Stupid*) mit einer Sequenz verknüpft werden. Spielifizierte oder pädagogische Elemente können über Annotationen an die einzelnen Game Moves angefügt werden (siehe Abbildung 62) (de Troyer et al., 2017, S. 578f.).

Bei der ATTAC-L handelt es sich um eine DSML, weshalb gemäß dem DSM-Ansatz eine Codegenerierung aus Modellen möglich und bei dieser Sprache auch vorgesehen ist. Darüber hinaus ist eine direkte Interpretation der Modelle aus dem ATTAC-L Editor heraus möglich, indem sie in einer vordefinierten und auf *Unity*-basierten 3D-Umgebung simuliert werden können (de Troyer et al., 2017, S. 595). Die Autoren dieser DSML geben an, dass ATTAC-L aktuell auf die Domäne des Cybermobbings reduziert ist, jedoch theoretisch auch auf andere Domänen adaptiert werden kann (de Troyer et al., 2017, S. 598). Diese Adaption ist bislang jedoch nicht erfolgt. Zudem sind nur ein paar wenige spielifizierte Elemente aus dem Konzept von Gamification mit der DSML abbildbar (z. B. Points, die als Score bezeichnet werden), was eine wesentliche Einschränkung der Expressivität in Bezug auf spielifizierte Lehre darstellt. ATTAC-L fokussiert zudem die Domäne der Serious Games und zielt

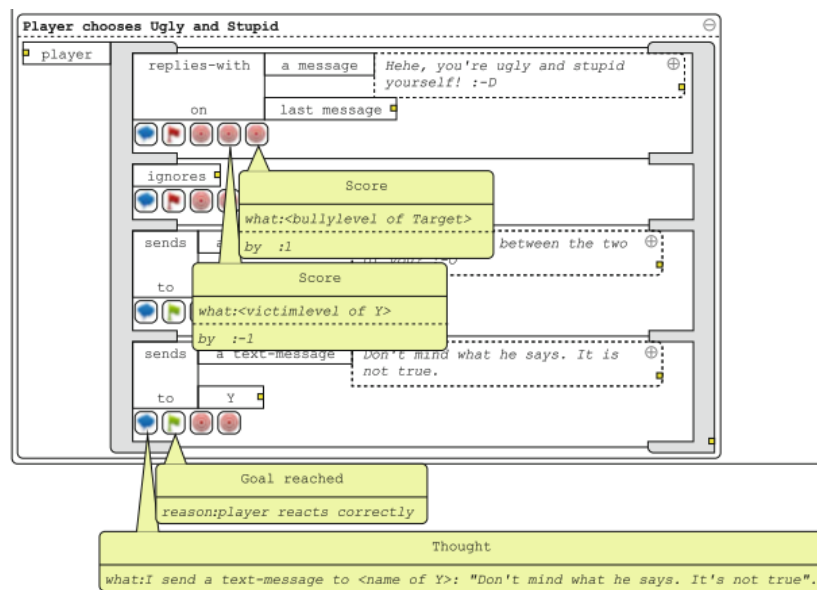


Abbildung 62: Auszug: Annotation eines Game Moves mit Scores (de Troyer et al., 2017, S. 578)

damit darauf ab, vollständige Spiele im Lehrkontext zu entwickeln, was von der Zielsetzung von Emendo abweicht. Weitere DSMLs, die im Kontext von Story-basierten Serious Games mit interaktivem Storytelling erstellt wurden, werden von de Troyer et al. (2017, S. 570) beispielhaft genannt und sollen hier ebenfalls Erwähnung finden, obwohl diesen Arbeiten mit der selben Argumentation wie bei ATTAC-L in dieser Arbeit keine weitere Beachtung geschenkt wird: *WEEV* (Marchiori et al., 2011), *Inform* (Nelson, 2006) und *GLiS-Mo* (Hirdes et al., 2012).

#### 6.6.6 Weitere relevante Arbeiten

Neben den vorgestellten domänenspezifischen Sprachen existieren noch weitere Arbeiten, die aufgrund ihrer technologischen Umsetzung und der jeweils einschlägigen Domänen Gamification und Hochschullehre (im Speziellen: Softwaremodellierung als Teildisziplin des Software Engineerings) hier im Folgenden Erwähnung finden sollen. Zudem sind sie von Interesse, da sie eine etwas andere Sichtweise auf spielifizierte Lehre haben und beispielsweise gänzlich auf eine Codegenerierung aus Modellen und anschließender Weiterverarbeitung in einer Zielumgebung verzichten und sich tendenziell eher die Fähigkeiten eines Modellierungswerkzeugs zu Nutze machen.

Eine beispielhafte Arbeit dafür stammt von Yohannis (2016). Er generiert auf Basis eines wohldefinierten Metamodells ein Spiel, das „[...] mimic a graphical modelling tool and at each level, it will require the learner to graphically construct or adapt a model to satisfy a set of requirements and constraints“ (Yohannis, 2016, S. 3). Die angesprochenen Anforderungen und Einschränkungen sind dabei beispielsweise in OCL definiert. Im Game werden sukzessive Level für Lernende freigeschaltet, die in Stories zusammengefasst sind und die gleichzeitig auch die Schwierigkeit der Modellierungsaufgaben erhöhen. Yohannis attestiert seinem Ansatz eine Flexibilität in Bezug auf die zu erlernende Modellierungssprache, die er aus Lehrendenperspektive auf Basis von GMF mit Hilfe des Frameworks *Eugenia* (Kolovos et al., 2017) je nach Bedarf definieren will. Aktuell ist ein Metamodell definiert, was eine vergleichsweise kleine Anzahl an Elementen unterstützt (flows, levels, challenges und objectives (Yohannis, 2016, S. 4)), ein auf Eclipse basierter grafischer Editor und ein Sketch eines Prototyps eines modellierten Games (siehe Abbildung 63).

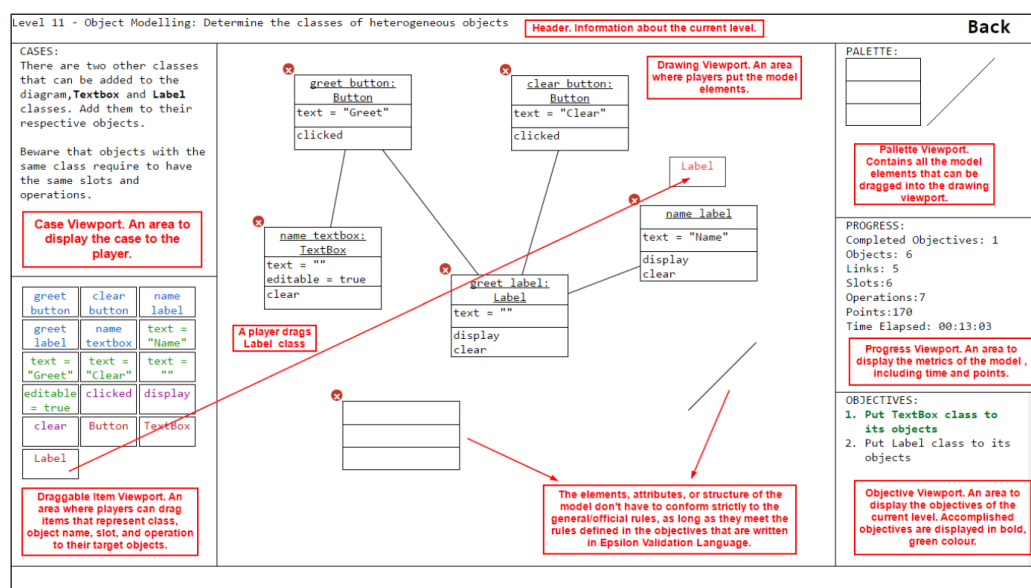


Abbildung 63: Sketch eines prototypischen spielifizierten Modellierungseditors (Yohannis, 2016, S. 5)

Einen vergleichbaren Ansatz, ebenso für das spielifizierte Lernen von Softwaremodellierung (im Speziellen UML und SQL), stellen Consentino et al. (2017) vor. Ihr Ansatz basiert auf einem Metamodell, welches unterteilt ist in ein Projekt-, Game- und Game-Status-Metamodell (Consentino et al., 2017, S. 17). Nur das Game Metamodell besitzt zur Spezifikation eine textuelle Syntax. Es

unterscheidet in die Game Design-Elemente *Level*, *Group*<sup>15</sup>, *Achievement*, wobei Level aus Groups bestehen und Groups bestimmte Themengebiete repräsentieren (Consentino et al., 2017, S. 18). Groups beinhalten Achievements, die nach erfolgreichem Abschließen von Tasks vergeben werden (Consentino et al., 2017, S. 18). Ob ein Task erfolgreich absolviert ist, bestimmt ein zuvor definierter OCL-Ausdruck (z. B. muss in einer Task eine UML-Klasse, was eine Ressource im Framework darstellt, mit einem bestimmten Namen von Lernenden modelliert werden, was mit OCL überprüft wird). Der Code Auszug 15<sup>16</sup> zeigt einen beispielhaften OCL-Ausdruck, der überprüft, ob eine UML-Klasse mit dem Namen *employee* in dem UML-Modell *myfirstmodel.uml* drei Attribute (*firstname*, *lastname* und *birthdate*) besitzt.

Code 15: OCL-Ausdruck zur Überprüfung von Modellierungsaktionen von Lernenden

```

1 -- Definitionen gekuerzt
2 files->select(f | f.name.toLowerCase() = 'myfirstmodel.uml' and f.model->
   notEmpty()).model->selectByKind(UML::Model).allowedElements()->
   selectByKind(UML::Class)
3 ->exists(c | c.name.toLowerCase() = 'employee' and
4   c.ownedAttribute->size() = 3 and
5   c.ownedAttribute->exists(p | p.name.toLowerCase() = 'firstname') and
6   c.ownedAttribute->exists(p | p.name.toLowerCase() = 'lastname') and
7   c.ownedAttribute->exists(p | p.name.toLowerCase() = 'birthdate')
8 )
9 -- Definitionen gekuerzt

```

Für das Bearbeiten von Tasks erhalten Lernende Rewards in Form von Points oder Badges, was ihren Game-Status verändert und Lernende mittels Notifications über diese Änderungen informiert werden (Consentino et al., 2017, S. 18, 21).

Beide hier genannten Beispiele, Modellierungseeditoren direkt für die Bearbeitung von Aufgaben durch Lernende zu verwenden, setzen eine von Emendo abweichende Art zu lernen um. Sie fokussieren ausschließlich das Gebiet der Softwaremodellierung durch (geleitete) Anwendungsaufgaben und stellen keine Wahlmöglichkeiten bei der Auswahl von Aufgaben zur Verfügung. Der Einsatz von weiteren Aufgabentypen ist jeweils nicht dokumentiert. Mo-

15 Nach Werbach u. Hunter (2015) handelt es sich bei einer Gruppierung nicht um ein GDE. Die Autoren fassen dies jedoch hier als ein GDE auf, obwohl eine Gruppe in diesem Fall den selben Character wie eine Container-Klasse in Emendo besitzt, von den Autoren jedoch nicht differenziert betrachtet wird.

16 Das Beispiel wurde auszugsweise entnommen aus <https://www.github.com/SOM-Research/gamification-modeling-learning/blob/master/codeskills.enduser/queries/queries.ocl>, abgerufen am: 01.06.2017.



dellierungseeditoren und darin eingebettete DSMLs sind nicht für eine Weiterverarbeitung durch Generatoren vorgesehen, sondern stellen direkt das Lernmittel dar. Die Spielifizierung beschränkt sich in beiden Fällen auf ein starres Set an Game Design-Elementen.

Emendo ist nicht auf das Gebiet der Softwaremodellierung beschränkt, sondern stellt eine nicht-fachgebundene Möglichkeit zur Verfügung, digitale Lehre flexibel durch Lernelemente und damit verbundenen Regeln zu spielifizieren. Aufgrund dieser unterschiedlichen Zielsetzung ist ein sinnvoller Vergleich dieser beiden Arbeiten mit Emendo, unter einheitlichen Kriterien, nur schwer zu vollziehen.

## 6.7 ZWISCHENFAZIT

Dieses Kapitel zeigte die Grundlagen von Technologien bzw. Frameworks, wie EMF und Sirius, mit denen die abstrakte und konkrete Syntax der Emendo DSML in ausführbare Formate überführt werden konnten. Es demonstrierte überblicksartig die GUI des Emendo Designers, ging auf relevante Aspekte beim Modellieren mit dem Emendo Designer ein und zeigte auf, wie dabei entstehende Herausforderungen gelöst werden konnten, mit dem Ziel, Nutzer bei der Erstellung von validen Modellen bestmöglich zu unterstützen. Zudem zeigte es die Integration des auf Acceleo basierenden Emendo Generators und ging auf die Struktur sowie die Anbindung des Generator Outputs an das im Folgenden zu beschreibende Domain Framework ein. Ebenso wurden auf Basis vorhandener Literatur Arbeiten präsentiert, die eine Ähnlichkeit zu Emendo aufweisen und diese Arbeit von den identifizierten Arbeiten abgegrenzt.

---

## EMENDO LEARNING MANAGEMENT SYSTEM

---

*„Always code as if the person who ends up maintaining  
your code will be a violent psychopath  
who knows where you live.“*

*— unbekannter Autor*

### 7.1 EINLEITUNG

Das Emendo LMS stellt das Domain Framework im Kontext des DSM-Ansatzes dar. Das LMS erfüllt aus Benutzersicht (sowohl Lehrende als auch Lernende) diverse Grundfunktionalitäten: Aus der Sicht von Lehrenden setzt es die zuvor im Emendo Designer modellierten und durch den Emendo Generator generierten, spielifizierten Lehr-Lernszenarien in eine für Lernende nutzbare Software automatisch um. Für Lehrende werden während der Nutzung der Plattform durch Lernende Fortschrittsmetriken erstellt und Möglichkeiten zum manuellen Feedback für zuvor modellierte Aufgaben gegeben. Lernende hingegen nutzen das LMS um sich spielifizierte Lerninhalte anzueignen bzw. zu bearbeiten, die von Lehrenden zuvor konzeptuell und inhaltlich für sie aufbereitet wurden. Zudem setzt das LMS weitere konzeptuelle Anforderungen um, die zuvor in dieser Arbeit definiert wurden, wie beispielsweise Diskussionsmöglichkeiten für Lernende (D\_A1–A6, D\_Z1) und integriert zudem spielifizierte Elemente, die weiter dazu beitragen, das erwartete Lernverhalten von Lernenden in Kombination mit bereitgestellten Lerninhalten, zu erzeugen (Ibanez et al., 2014).

### 7.2 ANFORDERUNGSANALYSE

Im Folgenden soll zunächst die Analyse der Anforderungen an das Emendo LMS im Fokus stehen. Sie wird anhand von strukturierten Dokumentationen mit der UML näher detailliert.

### 7.2.1 Funktionale und nichtfunktionale Anforderungen

Um einen systematischen Überblick über die Funktionalitäten des Emendo LMS zu erhalten und diese strukturiert umsetzen zu können, wurden funktionale und nichtfunktionale Anforderungen<sup>1</sup> an das LMS definiert. Eine vollständige Übersicht über die definierten Anforderungen, ist im Anhang A.5 dieser Arbeit zu finden. Die Formulierung der Anforderungen erfolgte auf Basis der Satzschablone nach Rupp (2009, S. 159ff.), welche in die Anforderungstypen

- *selbstständige Systemaktivität* (Bsp.: Das Emendo LMS muss Quellcode in Textblöcken formatiert anzeigen.),
- *Benutzerinteraktion* (Bsp.: Das Emendo LMS muss Lehrenden und Lernenden die Möglichkeit bieten, Diskussionsbeiträge zu Lernbausteinen zu kommentieren.) und
- *Schnittstellenanforderung* (in diesem Fall keine Interaktion mit Fremdsystemen, deshalb kein Beispiel für diesen Typ) unterteilt.

Außerdem wurde jede Anforderung auf Basis des Kano-Modells (Kano et al., 1984; Sauerwein, 2000) klassifiziert. Das Kano-Modell erlaubt eine Zuordnung jeder funktionalen Anforderung zu einer der drei folgenden Kategorien (diese wurden auf den Anwendungskontext des Emendo LMS adaptiert):

1. *Basisfaktoren* sind Anforderungen, die grundlegend und selbstverständlich für das Emendo LMS sind.
2. *Leistungsfaktoren* sind jene Anforderungen, die über die Basisfaktoren hinaus wichtige Features des Emendo LMS darstellen.
3. *Begeisterungsfaktoren* sind nützliche Features, die sich während der Konzeption und Entwicklung ergeben und die das Emendo LMS gegenüber anderen existierenden LMS positiv hervorheben.

Die Klassifizierung nach dem Kano-Modell ermöglicht eine Priorisierung der einzelnen Anforderungen für ein iterativ-inkrementelles Vorgehen (Rupp, 2009, S. 81), das der Konzeption und Entwicklung des Emendo LMS zu Grunde lag.

Während funktionale Anforderungen allgemein festlegen, was ein System tun bzw. können soll, beschreiben nichtfunktionale Anforderungen wie gut Funktionen umgesetzt werden sollen, in Form von Qualitätsanforderungen

---

<sup>1</sup> Zu Grunde gelegt ist das Begriffsverständnis beider Anforderungstypen von Rupp (2009).

oder unterspezifizierten funktionalen Anforderungen im Sinne von Einschränkungen (Balzert, 2009, S. 464f.; van Lamsweerde, 2013, S. 24).

Die nichtfunktionalen Anforderungen an das Emendo LMS wurden den Vorschlägen von Eeles et al. (2010, S. 158ff., 171ff.) folgend, erhoben. Sie befinden sich, wie die funktionalen Anforderungen auch, im Anhang (A.5) dieser Arbeit.

#### 7.2.2 *Use Cases des Emendo LMS*

Auf Basis der funktionalen Anforderungen wurden Use Cases definiert und ausformuliert. Die Abbildung 64 zeigt das entsprechende Use Case Diagramm mit den Akteuren Benutzer, Lernender und Lehrender.

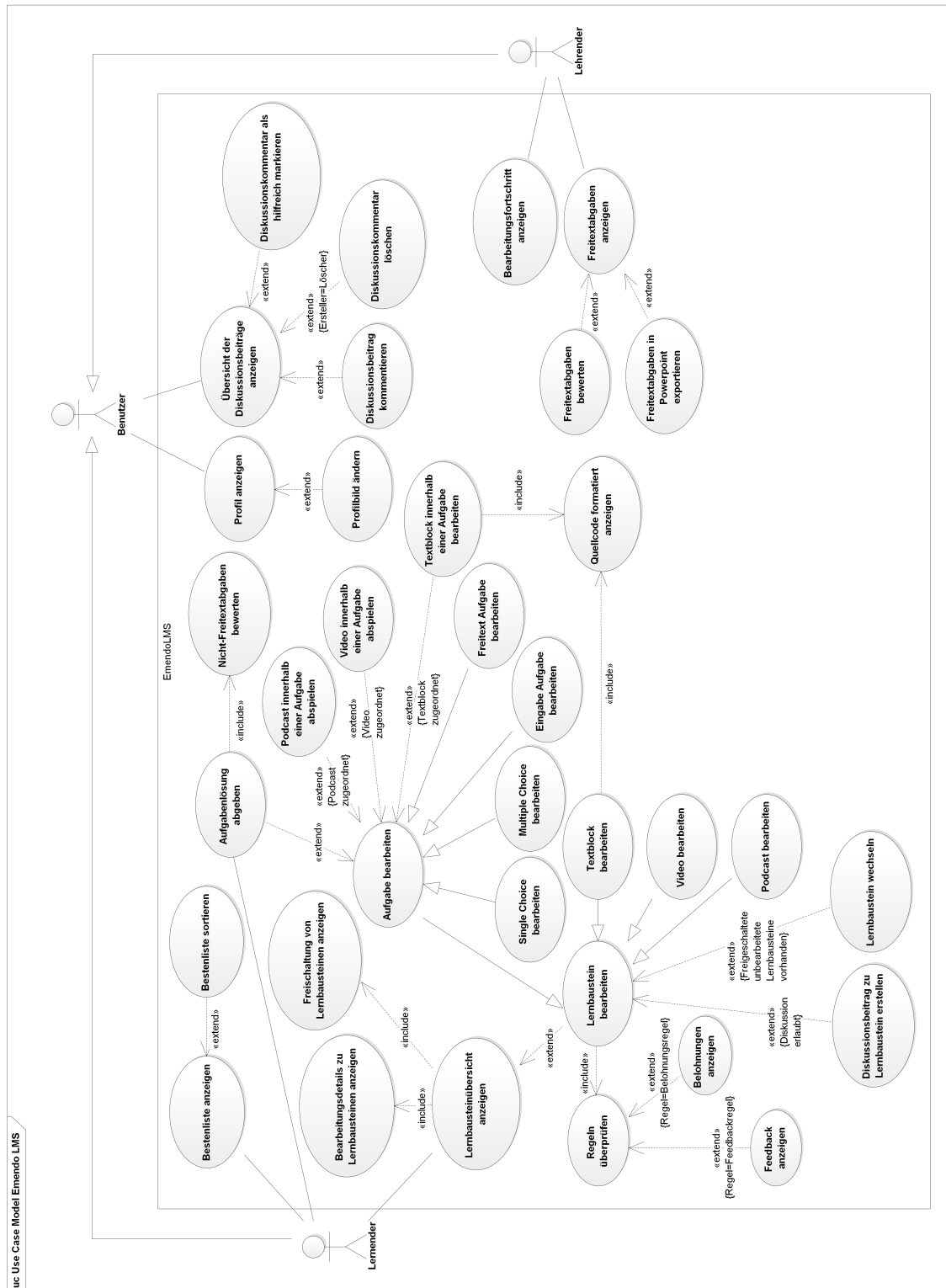


Abbildung 64: Use Case Diagramm Emendo LMS

Jeder Use Case wurde gemäß einem Schema, angelehnt an den Vorschlag von Rupp (2009, S. 215) beschrieben. Es besteht aus den Bestandteilen *Name*, *Kurzbeschreibung*, *Akteure*, *Vorbedingungen*, *fachlicher Auslöser* und *Essentielle Schritte im Normalfall*. Das Beschreibungsschema wurde um die Felder *Ergebnisse*, *eingehende Daten* und *Nachbedingungen* erweitert, um den Detaillierungsgrad der Use Cases zu erhöhen und die Fachlichkeit in einer Use Case Beschreibung besser herauszustellen. Die Tabelle 18 zeigt eine beispielhafte Beschreibung des Use Cases *Podcast bearbeiten*.

Name:	<b>Podcast bearbeiten</b>
Kurzbeschreibung:	Der Anwendungsfall beschreibt, welche Aktionen notwendig sind, um einen Podcast zu bearbeiten.
Akteure:	Lernender.
Vorbedingungen:	Der Lernende hat sich am Emendo LMS angemeldet. Die Detailansicht eines Lernbaustein Podcasts ist zur Bearbeitung geöffnet.
Fachlicher Auslöser:	Der Lernende startet den Podcast.
Ergebnisse:	Der Lernende hört den Podcast vollständig an. Der Podcast wird im LMS als bearbeitet markiert. Mit dem Podcast verknüpfte Regeln werden überprüft.
Eingehende Daten:	Ausgewählter Lernbaustein Podcast.
Nachbedingungen:	-
weiter auf nächster Seite	

Essentielle Schritte im Normalfall:	<ol style="list-style-type: none"> <li>1. Detailansicht zu Podcast ist geöffnet.</li> <li>2. Podcast wird durch Lernenden gestartet.</li> <li>3. Podcast stoppt, wenn vollständig angehört.</li> <li>4. Podcast im LMS als bearbeitet markieren.</li> <li>5. Dem Podcast zugeordnete Regeln überprüfen.</li> <li>6. Der nächste freigeschaltete Lernbaustein wird ermittelt.</li> </ol>
Ausnahmen, Fehlersituationen:	Fehlende Verbindung zum Internet; URL zum Podcast kann nicht korrekt aufgelöst werden; Podcast vorzeitig beendet; Podcastplayer nicht verfügbar.

Tabelle 18: Anwendungsfallbeschreibung Podcast bearbeiten

### 7.3 SYSTEMARCHITEKTUR

Auf Basis der durchgeführten Anforderungsanalyse und unter Berücksichtigung der zuvor erhobenen konzeptuellen Anforderungen, wurde die Systemarchitektur sukzessive und top-down entworfen. Das Ergebnis dieser Iterationen wird im Folgenden mit Hilfe von entsprechenden UML-Diagrammen beschrieben.

#### 7.3.1 *Model-View-Controller als Architekturmuster für das Emendo LMS*

Als grundlegendes Muster für Systeme wie das Emendo LMS, bei denen mit menschlichen Benutzern interagiert wird, bieten sich diverse interaktionsorientierte Muster an, die Interaktion und die dafür notwendige Logik strukturell trennen. Eines davon ist das Muster Model-View-Controller (MVC) (Busch-

mann et al., 2007, S. 188ff.; Starke, 2014, S. 156ff.; Geirhos, 2015, S. 519ff.). Gemäß dem SoC-Prinzip unterteilt MVC in die folgenden drei Rollen mit ihren jeweiligen Zuständigkeiten (Buschmann et al., 2007, S. 189; Geirhos, 2015, S. 521):

- Das Model steht für das *funktionale Herz* einer Anwendung, in dem es die Speicherung von Daten, samt aller Operationen, die auf Daten ausgeführt werden können, übernimmt. Es nutzt dabei eine oder mehrere Domänenobjekte (Buschmann et al., 2007, S. 208ff.).
- Die View stellt die Daten aus Domänenobjekten des Models dar. Dabei gilt, dass es zu jeder View genau ein Model gibt, jedoch zu einem Model mehrere Views, die die Darstellung der Daten übernehmen können. Dabei kann unterschieden werden in Template Views (Buschmann et al., 2007, S. 345), welche Daten aus dem Model schlichtweg für die Anzeige rendern und Transform Views (Buschmann et al., 2007, S. 347), die einzelne Modeldaten individuell und separat rendern.
- Der Controller stellt das Bindeglied zwischen View und Model dar. Er nimmt also die Eingaben von Benutzern in der Rolle eines Beobachters gemäß des Beobachter Musters (Gamma et al., 1995, S. 326ff.) entgegen, manipuliert daraufhin den Zustand des Models und sorgt für die Anzeige der geänderten Daten in der entsprechenden assoziierten View.

Da es sich bei MVC um ein Muster handelt, lässt es nicht nur Variation in der Anwendung zu, vielmehr fordert es eine Adaption auf den jeweiligen Anwendungskontext. Für das Emendo LMS wurde deshalb eine einheitliche Verwendung von MVC aufgrund der Anforderungen an das System festgelegt, was zu einer strukturierten Definition von Abhängigkeiten sowohl innerhalb eines MVC-Konglomerats als auch nach außen mit weiteren Komponenten zulässt. Im Kontext des Emendo LMS kennt ein Controller sowohl die View als auch das Model. Er sorgt dafür, dass die View mit den Daten aus dem Model initialisiert sowie die Funktionalität zur Ereignisverarbeitung (als Teil des Controllers) den Elementen der View hinzugefügt wird und meldet sich beim Model als Beobachter an, welches als konkretes Subjekt auftritt. Ändert sich der Zustand des Models, werden Beobachter darüber benachrichtigt, welche wiederum assoziierte Views aktualisieren. Zwischen View und Model besteht demnach beidseitig keine Beziehung, ebenso wenig wie zwischen verschiedenen Views oder verschiedenen Models. MVC-Triaden finden sich auf unterschiedlichen Abstraktionsstufen des Emendo LMS wieder. Sofern Funktionalität zwischen Controllern benötigt wird, sorgen Manager-Komponenten auf



den jeweiligen Abstraktionsstufen für einen einheitlichen Zugriff aufeinander, unter Einhaltung des Single Responsibility Principle (SRP) (Starke, 2014, S. 164). Dafür registriert sich jeder Controller bei dem Manager seiner Abstraktionsstufe bei Erzeugung.

### 7.3.2 *PanelHandler*

Grundsätzlich besteht das Emendo LMS aus drei Hauptkomponenten, dem *PanelHandler*, dem *LernelementHandler* sowie dem *RegelHandler*.

Der *PanelHandler* mit seinen Subkomponenten und Abhängigkeiten zu weiteren Komponenten des LMS ist in Abbildung 65 in einem Komponenten-diagramm dargestellt. Diese Komponente trägt vor allem die Verantwortlichkeit für die Erzeugung und korrekte Initialisierung der GUI unter Zuhilfenahme von diversen weiteren Komponenten.

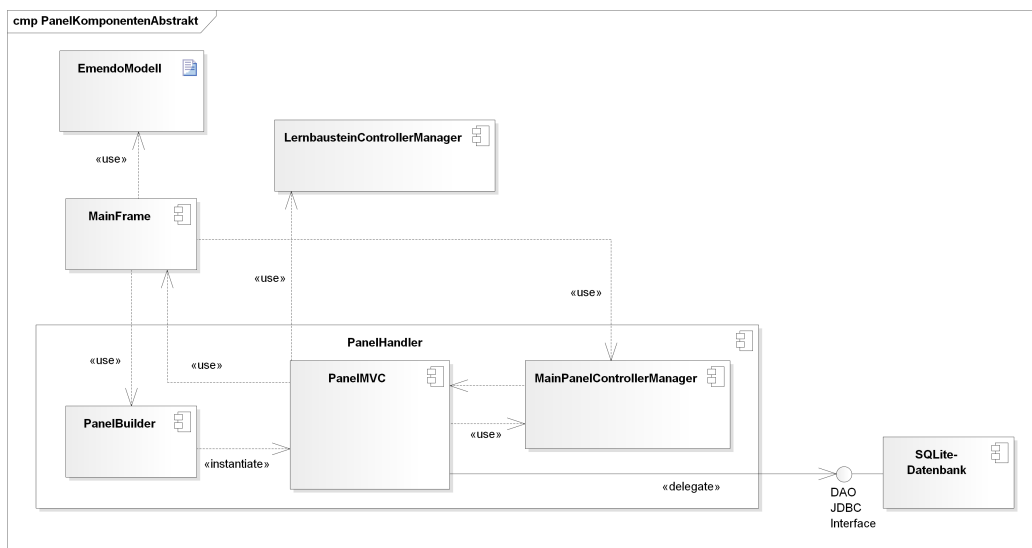


Abbildung 65: Die Komponente *PanelHandler*

Die Komponente *MainFrame* bildet das übergeordnete Root-Frame des LMS und gleichzeitig den Einstiegspunkt der Anwendung. Die Komponente ist global zugänglich und verwaltet das Hauptmenü und stellt Funktionalität bereit, um Views innerhalb der Anwendung sicher, also unter der persistenten Sicherung der jeweiligen Zustände von Views in der Datenbank, zu wechseln. Der Wechsel von Views vollzieht das *MainFrame* über die Komponente *MainPanelControllerManager*, welche den Zugriff auf alle Controller der Panel-MVCs auf dem *MainFrame* bereithält. Das *MainFrame* nutzt für das Erzeugen seiner Panel-MVCs den *PanelBuilder*, welcher die Panel-MVC-Triaden des *MainFrame*

mit Hilfe des Factory-Musters (Gamma et al., 1995, S. 121) erstellt. Die Komponente *MainFrame* stößt ebenfalls die Erstellung der Lernelemente an, die sich in dem durch den Emendo Generator generierten Java-Artefakt *EmendoModell.java* befinden.

### 7.3.3 *LernelementHandler*

Das Artefakt *EmendoModell.java* greift für die Erstellung seiner Elemente auf die Funktionalität zu, welche die Komponente *LernbausteinBuilder* bereitstellt und die als Subkomponente der zweiten Hauptkomponente, dem *LernelementHandler* in Abbildung 66 dargestellt ist.

#### 7.3.3.1 *Lernelement Erzeugung*

Im *LernbausteinBuilder* manifestiert sich das Muster Construction Builder (Fowler, 2010, S. 179ff.), das dafür sorgt, Domänenobjekte, die aus dem generierten Code der *EmendoModell*-Klasse stammen, korrekt zusammen zu bauen und anschließend zu instanziiieren. Deren Instanziierung sorgt gleichzeitig für die Erzeugung von weiteren MVC-Triaden, welche pro Lernbausteintyp angelegt werden und durch die Komponente *LernbausteinMVC* in Abbildung 66 repräsentiert sind. Deren Controller werden, wie bereits für die Hauptkomponente *PanelHandler* beschrieben, in der Komponente *LernbausteinControllerManager* auf dieser Abstraktionsebene verwaltet. Wenn gemäß der Definitionen im *EmendoModell* ein Lernelement, wie z. B. ein Textblock, als Teil einer Aufgabe (z. B. Single Choice) verknüpft ist, werden für das Lernmaterial lediglich ein *LernbausteinModel* und eine *LernbausteinWidgetView* instanziiert. Andernfalls werden zusätzlich noch eine *LernbausteinView* und ein *LernbausteinController* instanziiert, da es sich bei dem Lernmaterial um einen eigenständigen Lernbaustein handelt. Den Ablauf einer Instanziierung für den beispielhaften Fall, dass ein Textblock einer Single Choice-Aufgabe zugeordnet ist, dokumentiert das Sequenzdiagramm<sup>2</sup> aus Abbildung 67. Als Input dafür gelten lediglich die im *EmendoModell* verfügbaren Daten zu beiden Domänenobjekten und deren Abhängigkeiten zueinander.

In dem Sequenzdiagramm aus Abbildung 67 ist ebenfalls der Einsatz von Persistence Repositories (Millett u. Tune, 2015, S. 324; Fowler, 2003, S. 322) dokumentiert. Dabei handelt es sich um Komponenten, die für Create-Read-Update-Delete (CRUD)-Aktionen eines jeweilig (zusammengesetzten) Domä-

<sup>2</sup> Aus Gründen der Übersichtlichkeit wurde darauf verzichtet, die Gruppe der Lernbausteine Textblock und Single Choice-Aufgabe sowie mögliche mit beiden Elementen assoziierte Regeln im Sequenzdiagramm ebenfalls darzustellen.

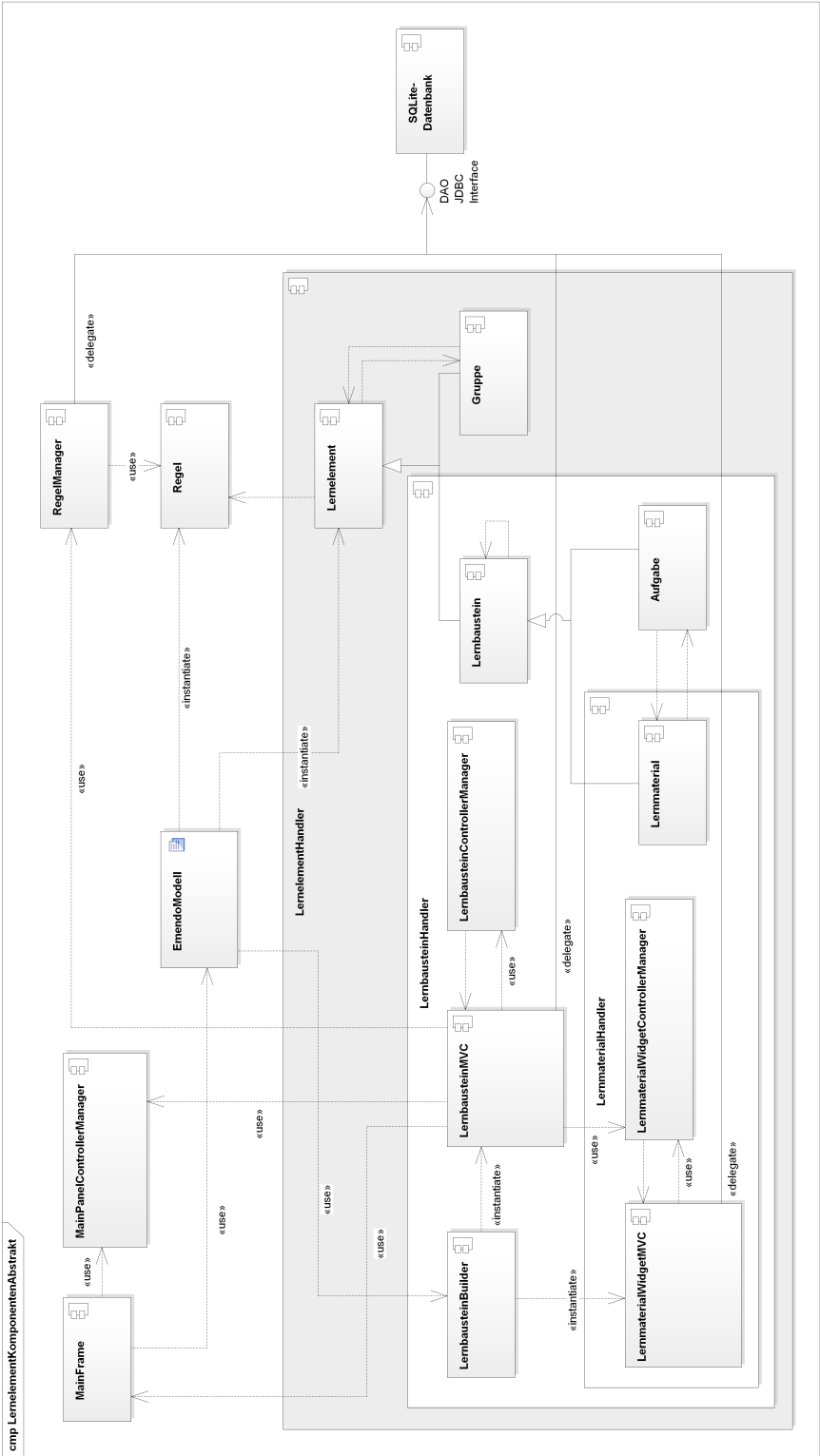


Abbildung 66: Die Komponente LernelementHandler

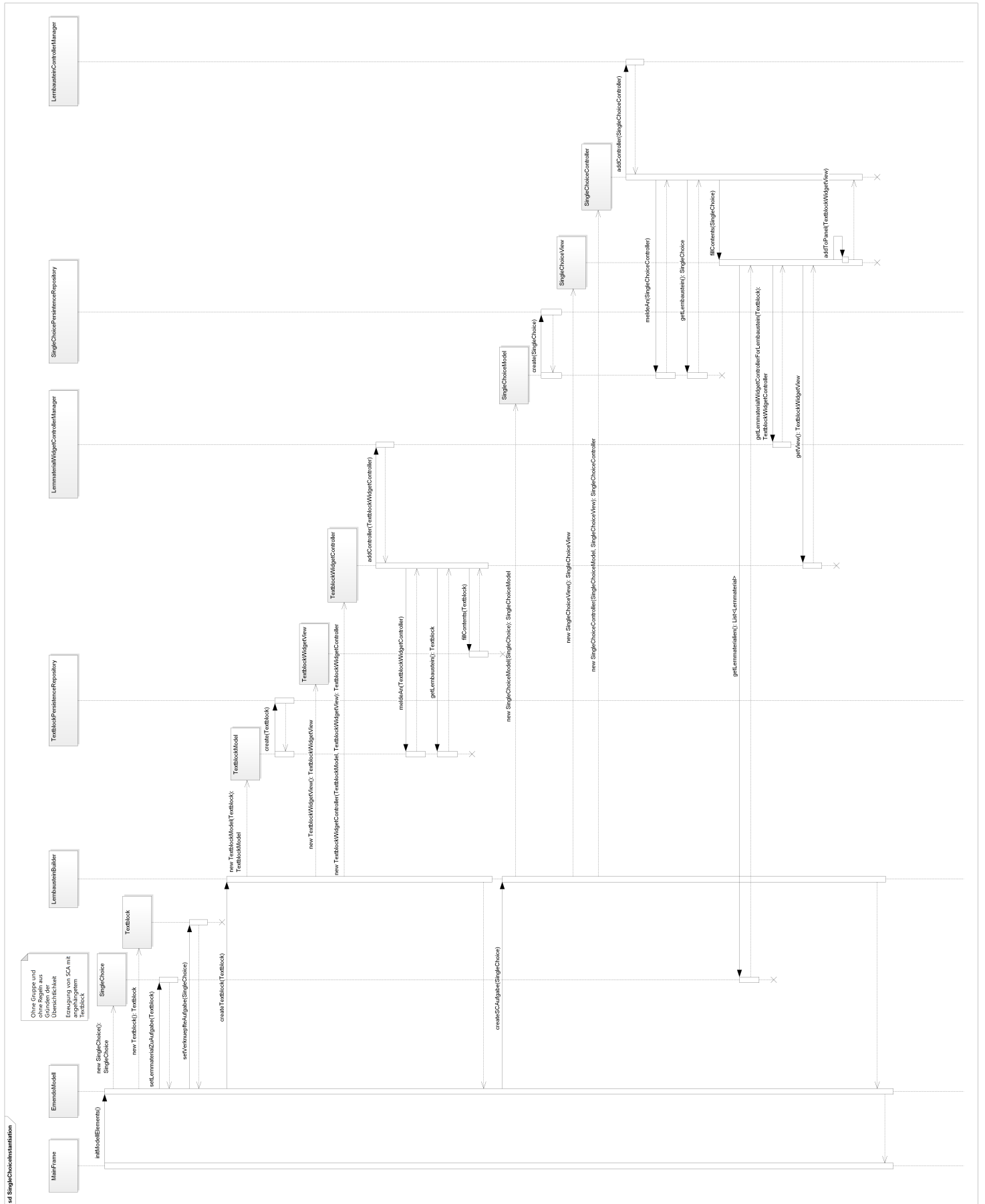


Abbildung 67: Sequenzdiagramm - Erstellung einer Single Choice-Aufgabe gemäß dem Construction Builder-Muster

nenobjekts zuständig sind und damit die fachlichen Domänenobjekte von ihrer Persistierung bzw. ihrem Laden trennen. Auf die Datenhaltung soll später noch genauer eingegangen werden.

#### 7.3.3.2 Komponente *Lernelement*

Eine wesentliche Subkomponente des *LernelementHandlers* stellt die Komponente *Lernelement* dar. Sie ist mit ihren Subkomponenten in Abbildung 70 dargestellt. Hier findet sich eine vergleichbare Struktur zum Metamodell der Emendo DSML wieder, nämlich die Spezialisierung der Komponente *Lernelement* in die beiden Komponenten *Lernbaustein* und *Gruppe*, wobei *Lernbaustein* wiederum eine Generalisierung von *Lernmaterial* und *Aufgabe* darstellt. Ebenso wie im Metamodell definiert, können Regeln *Lernelementen* zugeordnet sein. Außerdem ist jedes *Lernelement* Bestandteil einer *Gruppe*, ausgenommen der *Wurzelgruppe*, welche selbst eine *Gruppe* darstellt und die alle Elemente eines Modells auf oberster Ebene beinhaltet.

Darüber hinaus lässt sich die Komponente *Lernbaustein* in die bereits aus dem Metamodell der Emendo DSML bekannten Subkomponenten *Single Choice*, *Multiple Choice*, *Freitext* und *Eingabe* unterteilen, welche allesamt Aufgaben darstellen sowie Lernmaterialien, wie *Podcast*, *Video* und *Textblock*, welche entweder eigenständig oder eingebettet in eine Aufgabe existieren können. Diese strukturelle Nähe zum Metamodell erlaubt eine effiziente Nutzung und Weiterverarbeitung des Outputs des Emendo Generators im Emendo LMS. Die *Textblock*-Komponente (bzw. auch die Komponente *Aufgabe*, sofern ein *Textblock* Bestandteil einer Aufgabe ist) nutzt zudem die Bibliothek *RSyntax-TextArea.jar*, welche für eine formatierte Anzeige von Quellcode innerhalb eines *Textblocks* sorgt.

Alle Subkomponenten von *Aufgabe* und *Lernmaterial* nutzen die bereits angesprochenen Persistence Repositories, um ihre jeweiligen Zustände in der Datenbank zu sichern und entsprechend bei Aufruf zu laden. Dabei durchlaufen die konkreten Instanzen der Komponente *Aufgabe* die in Abbildung 68 dargestellten Zustände. Es handelt sich bei den Zuständen um jene, die für alle Typen von Aufgaben Gültigkeit besitzen und deshalb keine weitere Differenzierung anhand des konkreten Aufgaben-Typs notwendig ist.

Anders verhält es sich bei den Zuständen von eigenständigen und nicht Aufgaben zugeordneten Lernmaterialien. Sie sind ebenfalls in einem Zustandsdiagramm (siehe Abbildung 69) dokumentiert. Sind Lernmaterialien Aufgaben zugeordnet, überwiegen die Zustände von Aufgaben und die von Lernmaterialien werden ignoriert.

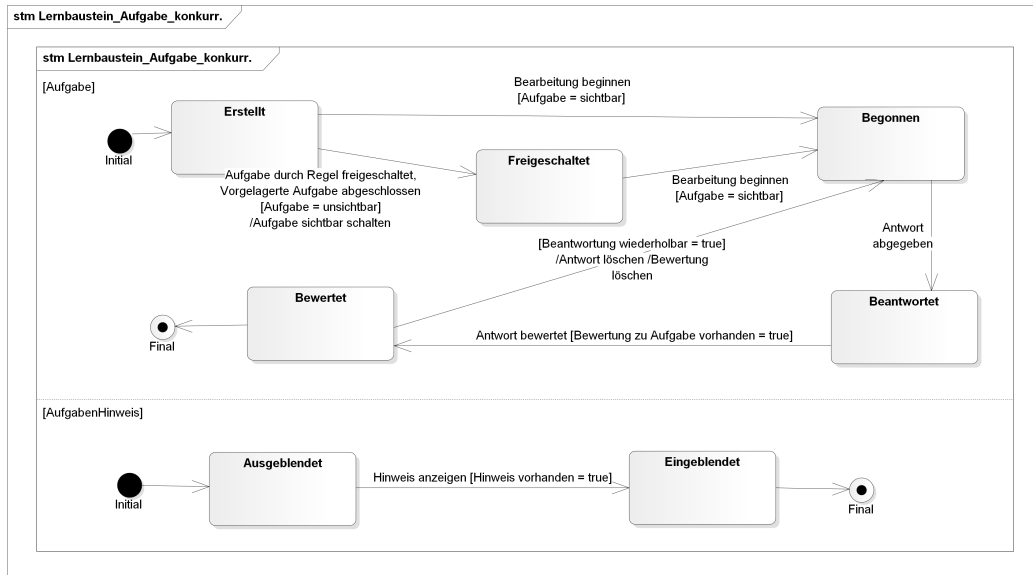


Abbildung 68: Zustandsdiagramm Aufgabe und Aufgabenhinweis

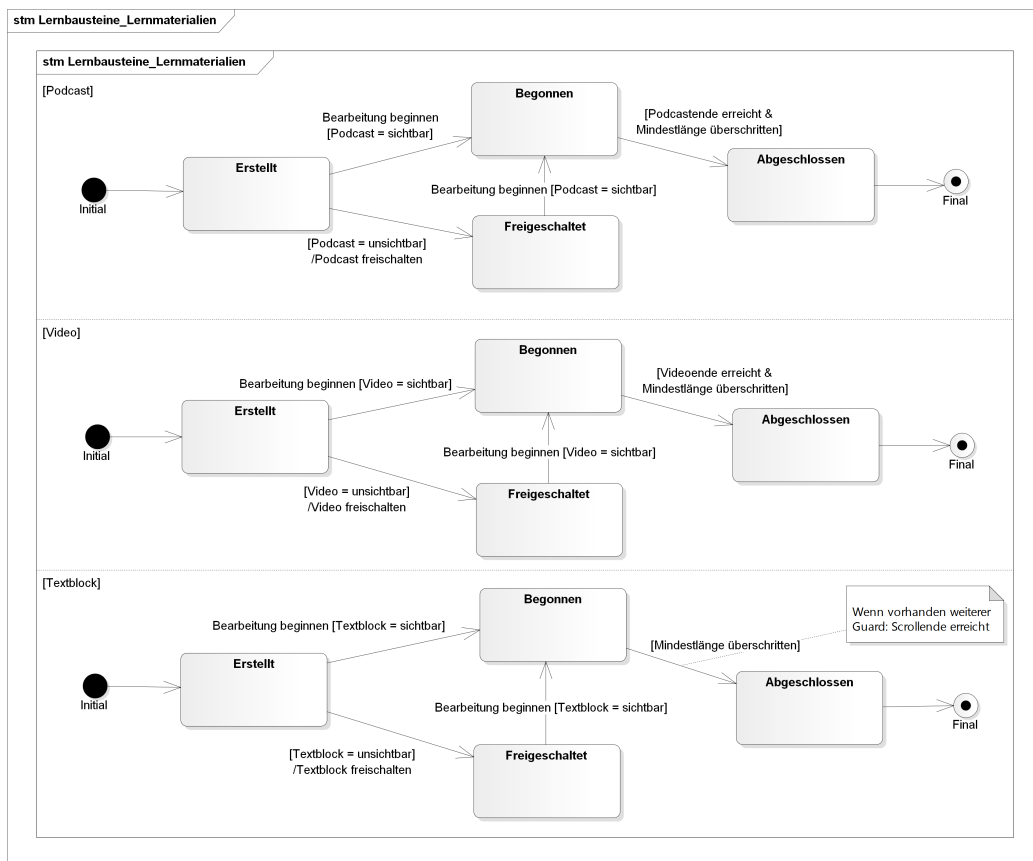


Abbildung 69: Zustandsdiagramm für die Lernmaterialien Podcast, Video und Textblock

Obwohl die Zustände der jeweiligen Lernmaterial-Typen auf einen ersten Blick gleich erscheinen, so unterscheiden sie sich doch maßgeblich in den Ak-

tionen oder Bedingungen, die zu Zustandsübergängen führen. Die für die Zustandsübergänge notwendige Funktionalität ist demnach individuell für die jeweiligen konkreten Lernmaterial-Typen.

Darüber hinaus nutzt die Komponente Lernbaustein die Komponente *Emendo Notification Manager*, über die alle Benutzerbenachrichtigungen bei Zustandsänderungen von Lernbausteinen, beispielsweise für den Erhalt eines Badges oder den Anzeigen eines Hinweises zu einer Aufgabe, abgewickelt werden.

Wenn es zu Zustandsänderungen von Lernbausteinen kommt, werden zudem mit Lernbausteinen assoziierte Regeln überprüft. Regeln sind durch die gleichnamige Komponente realisiert. Diese beinhaltet zudem eine dem Metamodell der DSML ähnliche Strukturierung, welche im Folgenden als Teil der dritten Hauptkomponente *RegelHandler* näher detailliert werden soll.

#### 7.3.4 *RegelHandler*

Die dritte Hauptkomponente *RegelHandler*, dargestellt in Abbildung 71, kümmert sich um die gesamte Ausführung von Regeln, die Lernelementen zugeordnet werden können.

Das Artefakt *EmendoModell.java* sorgt zunächst dafür, dass die mit Lernelementen assoziierten Regeln im Emendo LMS zusammengebaut und instanziiert werden. Dafür müssen die Bestandteile einer Komponente *Regel* erstellt werden, bevor eine vollständige Regel korrekt instanziiert werden kann. Konkret wird zuerst die *Benutzeraktion* der Regel erstellt und ihr hinzugefügt werden, bevor alle Ausdrücke geparkt, instanziiert und ebenfalls mit der Regel assoziiert werden. Schließlich werden die konkreten Konsequenzen erstellt, ihnen die assoziierten Belohnungen (*GameDesignElemente*) oder *Feedback*-Objekte hinzugefügt und abschließend alle Bestandteile in den *RegelBuilder* gegeben, der über eine Factory (Gamma et al., 1995, S. 121ff.) eine Regelinstanz zurückliefert, die einem Lernelement angehängt wird.

Auch in diesem Fall wurde sich für eine strukturelle Ähnlichkeit zum Metamodell der DSML bewusst entschieden (wie am Aufbau der Komponente zu erkennen), ebenfalls mit dem Ziel, den Aufwand für das Mapping zwischen Generator Output und Emendo LMS möglichst gering zu halten.

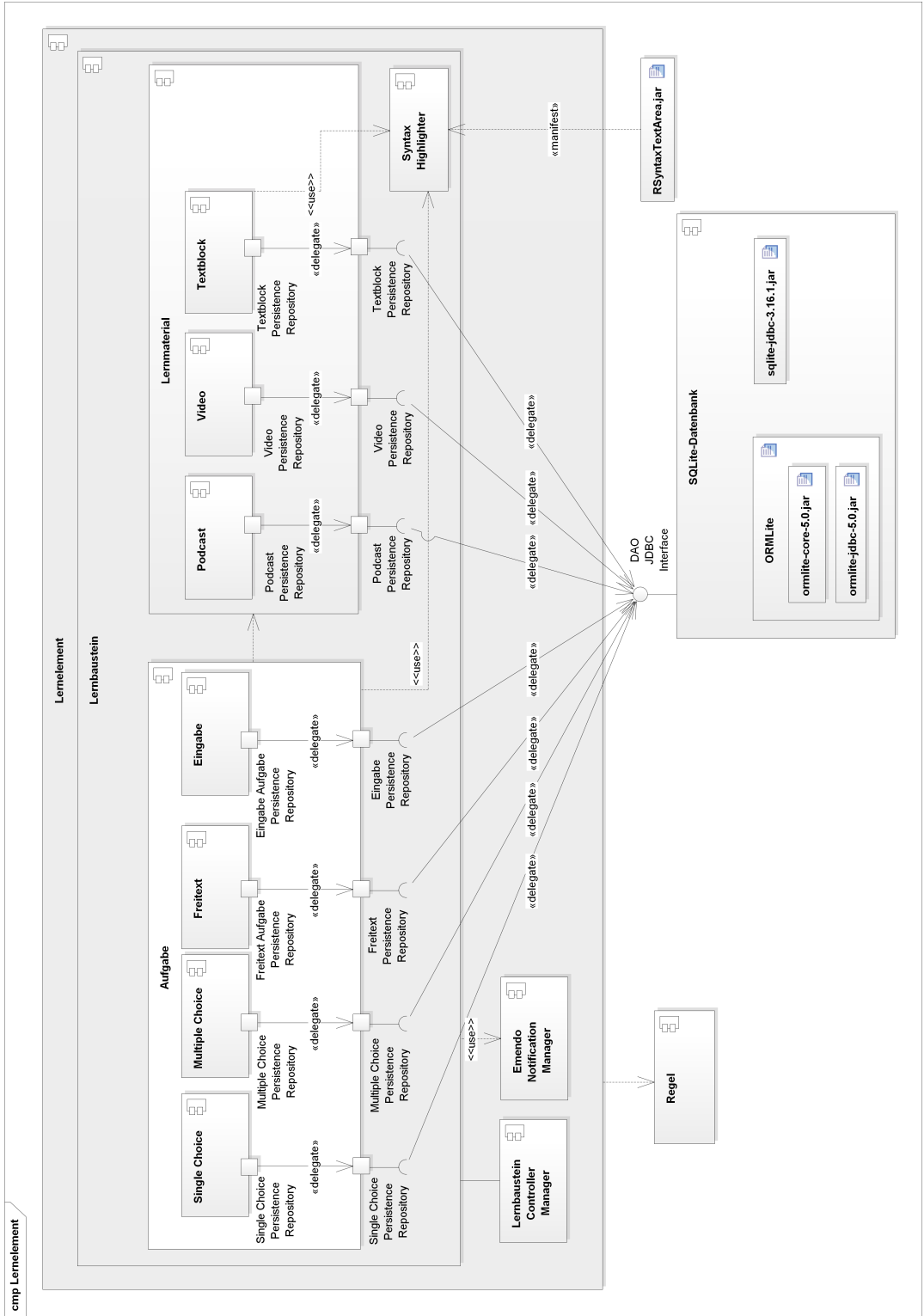


Abbildung 70: Die Komponente Lernelement



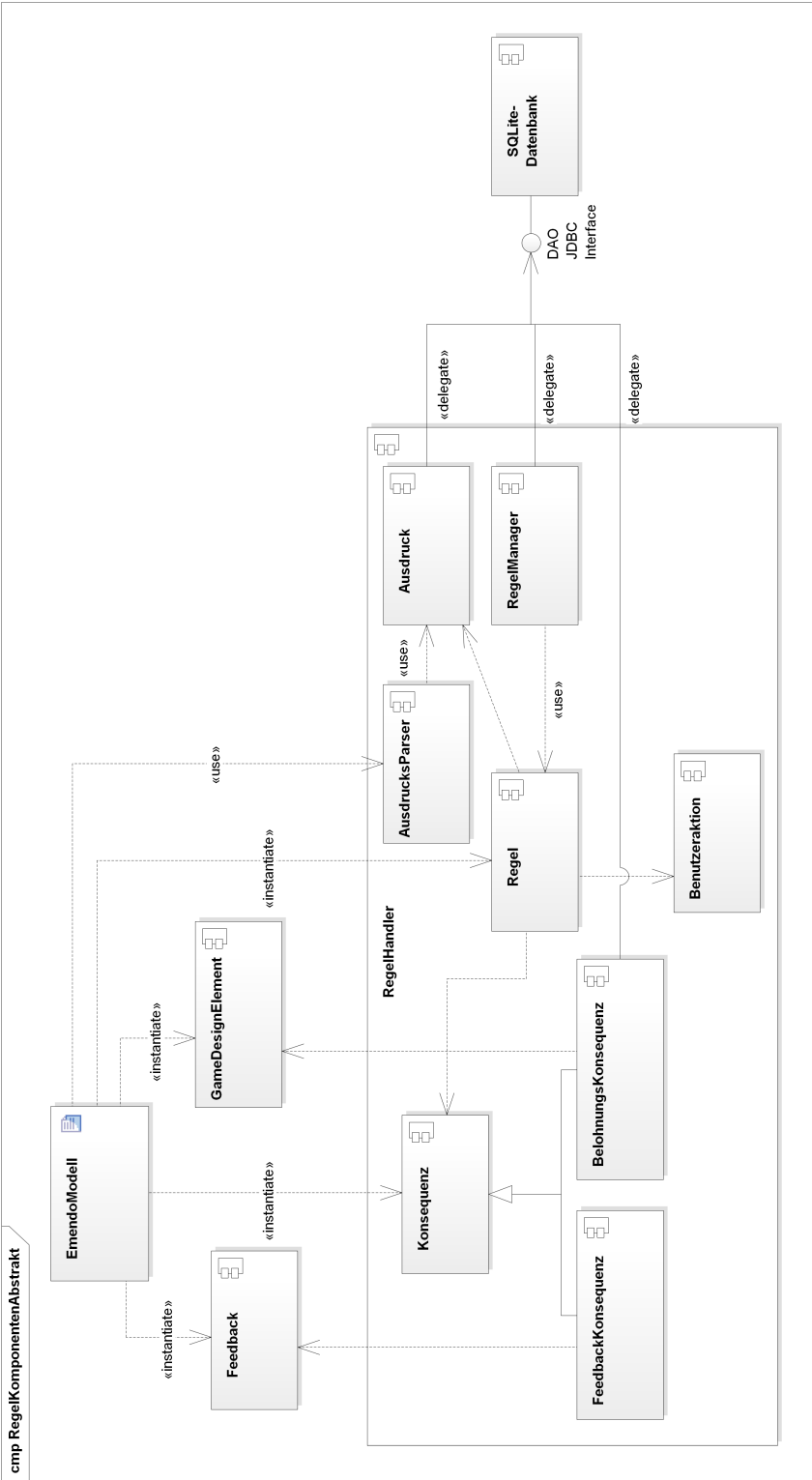


Abbildung 71: Die Komponente RegelHandler

### 7.3.4.1 Regelparsing

Ein besonderes Augenmerk soll auf dem Parsing eines Regelausdrucks durch die Komponente *AusdrucksParser* liegen. Diese sorgt dafür, dass Ausdrücke, die als Bedingungen der Regeln fungieren durch das Emendo LMS korrekt konstruiert werden können. Die Vorarbeit hierfür wird bereits im ecore-Meta-modell der DSML geleistet. Dort ist definiert, dass jede Regel eine *EString*-Liste an Ausdrücken hält. Jeder Ausdruck wird durch die entsprechenden Eingabe-Dialoge des Emendo Designers geklammert und mit Leerzeichen innerhalb eines Klammerpaares versehen und in dieser Form auf Basis von ecore-Objekten in Zeichenketten serialisiert. Die dabei exemplarisch entstehende Repräsentation auf Datenebene innerhalb einer Modellinstanz von einer Regel, wird in Abbildung 72 veranschaulicht (die logischen Und-Operatoren fungieren zu Darstellungszwecken als Trennzeichen der einzelnen Listenelemente und deuten an, dass die Ausdrücke bei ihrer Auswertung miteinander verundet werden).

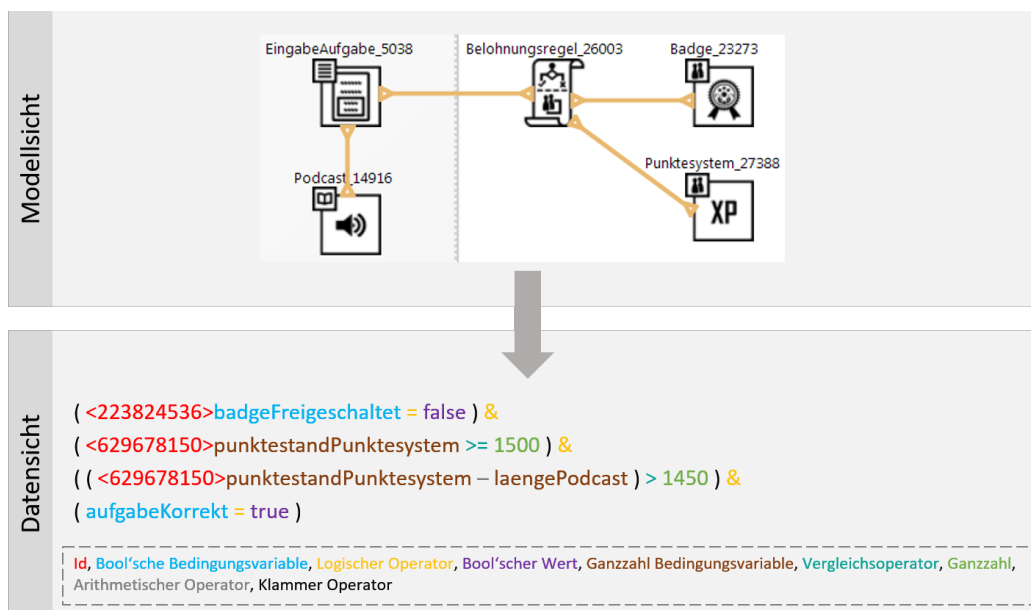


Abbildung 72: Interne Datenrepräsentation von vier Regel-Ausdrücken in einer String-Liste einer Emendo Modell Instanz inkl. Legende

Das Beispiel in Abbildung 72 zeigt nur einen Ausschnitt<sup>3</sup> aus dem in der Komponente *Ausdruck* verfügbarem Repertoire an Datentypen, die in einem *EString*-Ausdruck vorkommen können und von dem *AusdrucksParser* geparkt

<sup>3</sup> Natürlich besitzen die dargestellten Domänenobjekte, wie die *Eingabe Aufgabe* und der *Podcast*, ebenso eine interne Datenrepräsentation. Der Fokus ist soll hier jedoch auf die Regeln gelegt werden, weshalb die Sicht auf die Daten außerhalb dieses Scopes an dieser Stelle keine Beachtung findet.

und damit wieder in Objekte des LMS deserialisiert werden. Um einen Ausdruck zu parsen, teilt der Parser diesen zunächst anhand der darin vorkommenden Leerzeichen in Tokens auf. Die dabei entstehende Liste aus Strings stößt den Parsing-Vorgang an, der in Abbildung 73 durch ein Aktivitätsdiagramm auf oberster Ebene beschrieben ist.

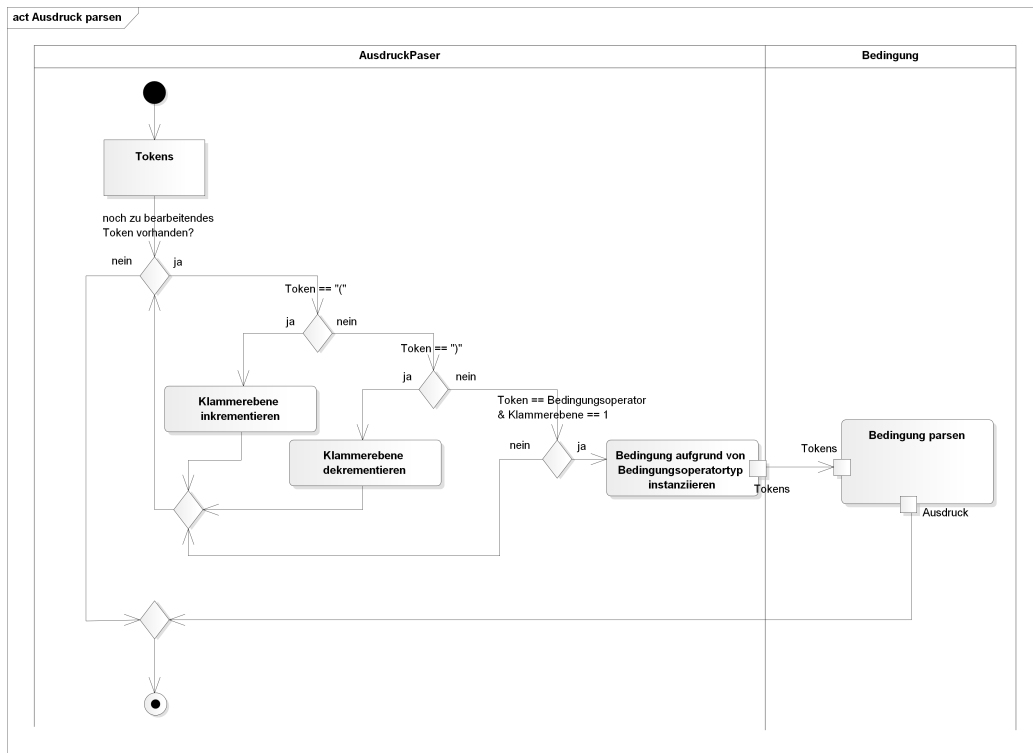


Abbildung 73: Aktivitätsdiagramm - Ausdruck parsen

Das bereits vorgestellte Prinzip der Klammerung eines jeden Ausdrucks, erlaubt ein rekursives Parsen von Bedingungen, beginnend nach positiver Überprüfung ob es sich um einen Ausdruck handelt, welcher durch eine Klammer geöffnet aber noch nicht wieder geschlossen wurde ( $\text{Klammerebene} == 1$ ) und es sich bei dem aktuell geprüften Token der Liste um einen Bedingungsoperator handelt, der als *Bedingungstyp*<sup>4</sup> im LMS verfügbar ist. Ist dies der Fall, wird anhand des Bedingungsoperators eine Instanz des Bedingungstyps erzeugt und in die Aktivität *Bedingung parsen* gesprungen, die in einem separaten Aktivitätsdiagramm in Abbildung 74 dargestellt ist.

In der Aktivität *Bedingung parsen* wird zunächst der linke Operand der Bedingung ermittelt. Hierfür wird geprüft, ob es sich bei dem Token links neben

<sup>4</sup> Die Bedingungstypen sind separate Objekte, die im Vorfeld bekannt sind und die das Parsen eines Ausdrucks individuell vornehmen. Jeder Bedingungstyp wird vor dem Parsing in der Komponente *RegelHandler* in einer statischen Singleton-Klasse (Gamma et al., 1995, S. 144ff.) *BedingungenManager* registriert.

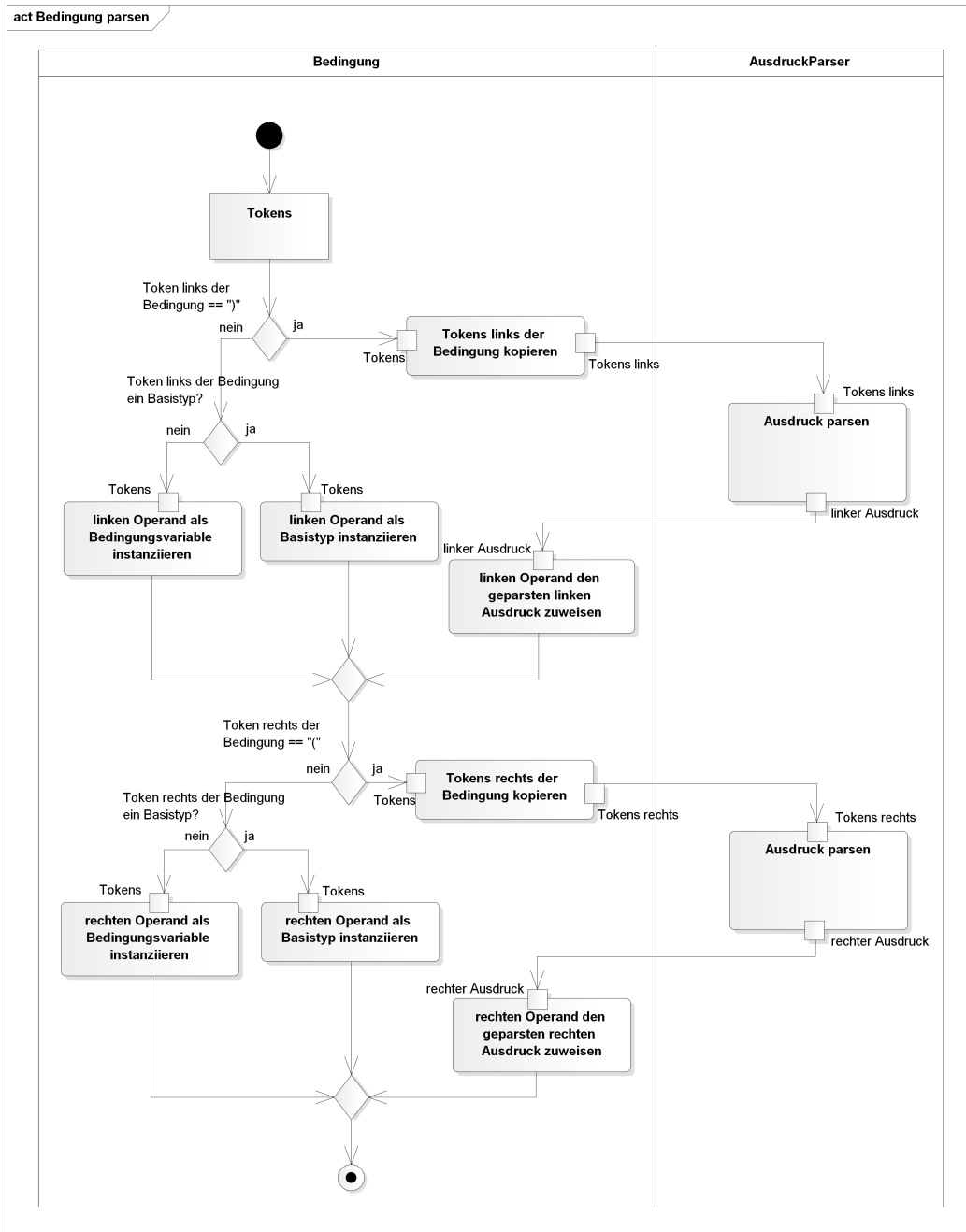


Abbildung 74: Aktivitätsdiagramm - Bedingung parsen

dem Operator, um eine schließende Klammer handelt. Falls ja, handelt es sich beim linken Operanden der Bedingung, um einen komplexen Ausdruck (ein Ausdruck, der aus weiteren Ausdrücken besteht), wofür weiteres rekursives Parsing notwendig ist. Dafür werden alle Tokens links des Bedingungsoperators kopiert, um ein Überschreiben bei einem komplexen Ausdruck zu vermeiden und die Aktivität *Ausdruck parsen* wird erneut mit den Tokens links vom Bedingungsoperator aufgerufen. Daraus resultiert ein *Ausdruck*-Objekt,

welches bei der Bedingung als linker Operand gesetzt wird. Handelt es sich bei dem zu Beginn dieser Aktivität überprüften Token nicht um eine schließende Klammer, wird überprüft, ob es sich um ein Basistyp (z.B. Integer oder String) oder eine Bedingungsvariable handelt. Handelt es sich um einen Basistyp, wird der Token als Basistyp über eine Factory (Gamma et al., 1995, S. 121ff.) instanziiert und als linker Operand der Bedingung gesetzt. Falls eine Bedingungsvariable (z. B. *AufgabeKorrekt* oder *AnzahlWiederholungen*) vorliegt, wird mit dem Namen der Bedingungsvariable als Parameter der Java-Classloader zu dieser Klasse aufgerufen und anschließend ein Objekt dieser Klasse erzeugt, das wiederum dem linken Operanden der Bedingung zugewiesen wird. Durch eine dieser drei beschriebenen Möglichkeiten ist damit der linke Operand der Bedingung festgelegt und selbiges Procedere wird für die Festlegung des rechten Operanden der Bedingung durchgeführt. Sind sowohl linker Operand als auch rechter Operand bestimmt (der Operator dazwischen ist ja bereits seit Beginn der Aktivität *Ausdruck parsen* bekannt), ist die Bedingung vollständig geparst und wird dem RegelBuilder für die finale Erstellung der Regel hinzugefügt. Anschließend wird mit noch nicht geparsten Ausdrücken, wie beschrieben, verfahren.

#### 7.3.4.2 Regelauswertung

Regeln werden durch Benutzeraktionen auf Lernbausteinen getriggert. Das Triggern sorgt dafür, dass die Komponente *RegelManager* eine Überprüfung von Regeln anstößt, die mit dem Lernbaustein assoziiert sind, welcher den Trigger verursacht hat. Der Prozess, den der *RegelManager* in Kooperation mit den Komponenten *Regel* und *Lernbaustein* dabei durchläuft, ist in Abbildung 75 dargestellt.

Nach initialen Überprüfungen, ob Regeln bestehen, deren Benutzeraktionen mit denen übereinstimmen, die das Triggern zur Überprüfung verursacht haben, wird in die Aktivität *Regel evaluieren* gesprungen, die in Abbildung 76 abgebildet ist.

In der Aktivität *Regel evaluieren* werden alle Ausdrücke sukzessive interpretiert. Als Eingabeparameter dieser Aktivität sind die Objekte des auslösenden Lernbausteins und die Gruppe in der er sich befindet vorhanden. Für die Interpretation der Ausdrücke, die in der Rolle von Bedingungen hier auftreten, wurde auf das Interpreter Muster (Gamma et al., 1995, S. 274ff.) zurückgegriffen. Die grundlegende Idee bei der Anwendung dieses Musters ist, dass jeder Ausdruck selbst weiß, wie er in einem bestimmten Kontext (in diesem Fall definiert durch den triggernden Lernbaustein und die Gruppe in der dieser sich befindet) zu interpretieren ist. Die Voraussetzung hierfür wurde durch das

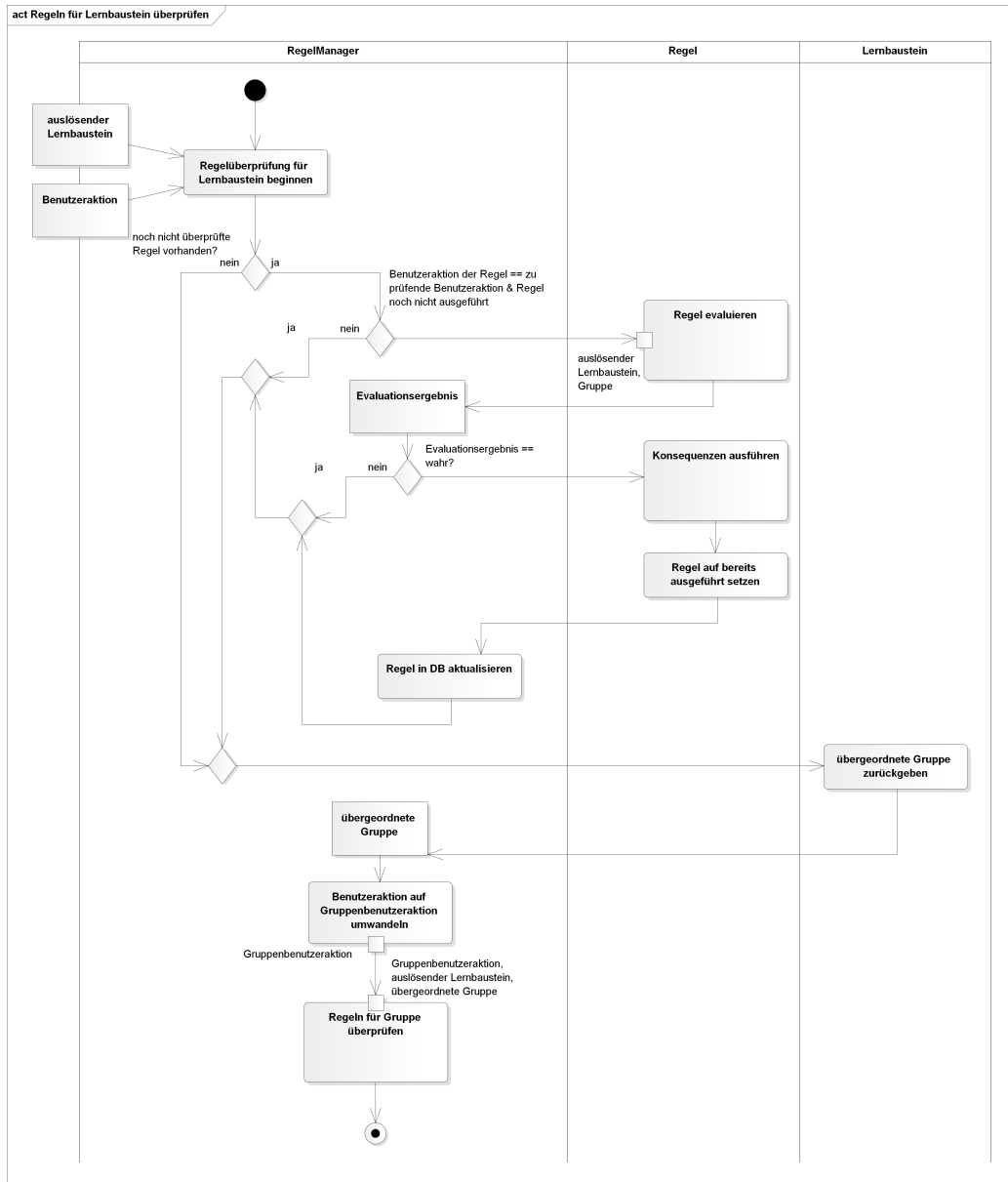


Abbildung 75: Aktivitätsdiagramm - Regeln für Lernbaustein überprüfen

Parsing von Bedingungen geschaffen, indem – ähnlich wie in der rekursiven Struktur des Kompositum Musters (Gamma et al., 1995, S. 183ff.) – ein abstrakter Ausdruck in einfache und komplexe Ausdruckstypen unterschieden werden kann. Einfache Ausdruckstypen stellen Basistypen oder Bedingungsvariablen dar, während komplexe Ausdruckstypen aus weiteren (abstrakten) Ausdrücken zusammengesetzt sind. Wird nun das Interpretieren einer Bedingung angestoßen (siehe Abbildung 77), werden ähnlich wie beim bereits beschriebenen Parsing, die Bestandteile der Bedingung interpretiert, beginnend beim linken Operanden.

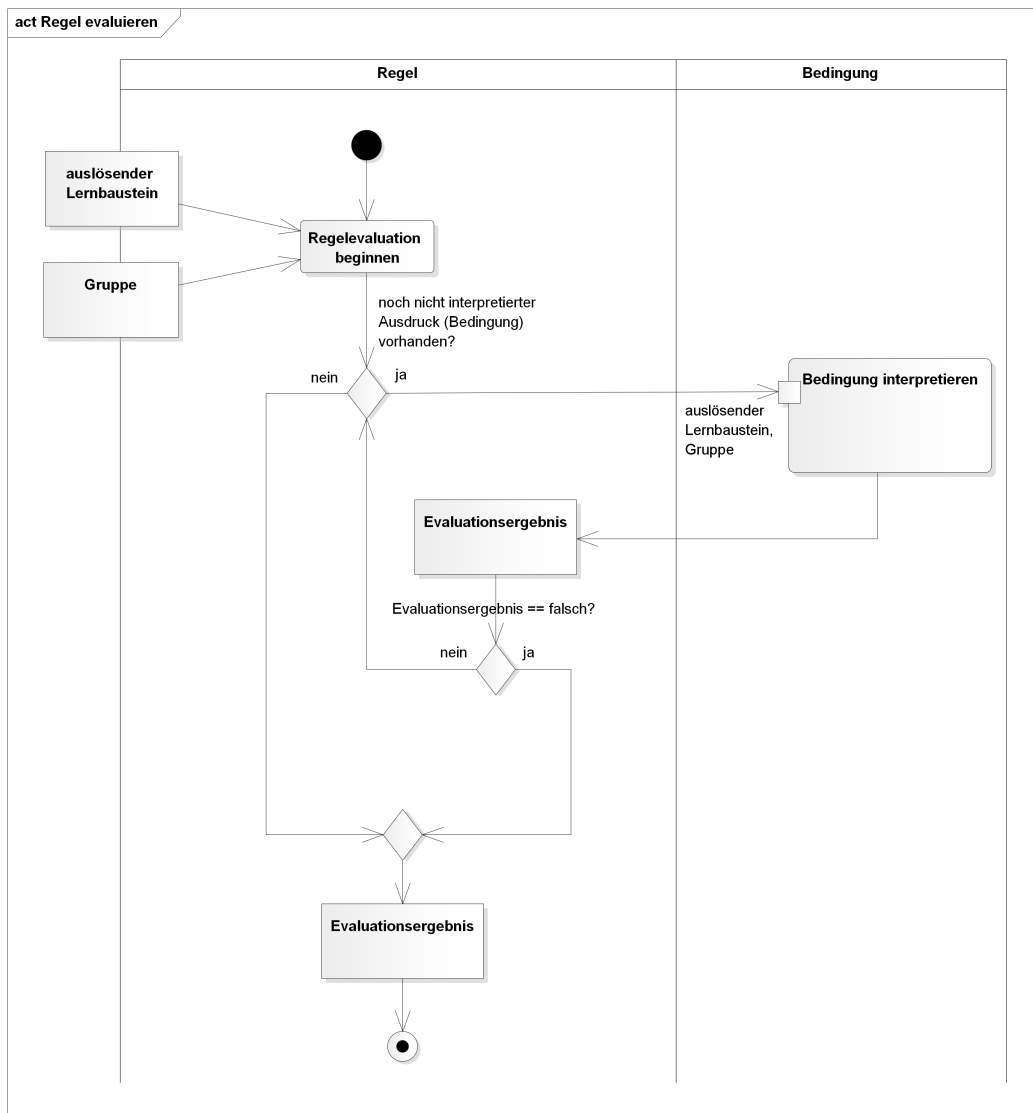


Abbildung 76: Aktivitätsdiagramm - Regel evaluieren

Zunächst wird überprüft, ob der linke Operand der Bedingung ein Basistyp ist und in diesem Fall, der Wert des Basistyps an den Aufrufer zurückgegeben und als linker Operand gesetzt. Ist dies nicht der Fall, wird überprüft, ob es sich bei dem linken Operanden um eine Bedingungsvariable handelt. Ist diese Überprüfung positiv ermittelt die Bedingungsvariable ihren Wert für den gegebenen Kontext, beispielsweise durch eine Abfrage auf die Datenbank und gibt diesen an den Aufrufer zurück, welcher den Wert als linken Operanden setzt. Schlagen beide Überprüfungen fehl, also handelt es sich bei dem linken Operanden nicht um einen Basistyp oder eine Bedingungsvariable, dann liegt ein komplexer Ausdruck vor, welcher mindestens einen weiteren rekursiven Aufruf der Aktivität *Bedingung interpretieren* erfordert.

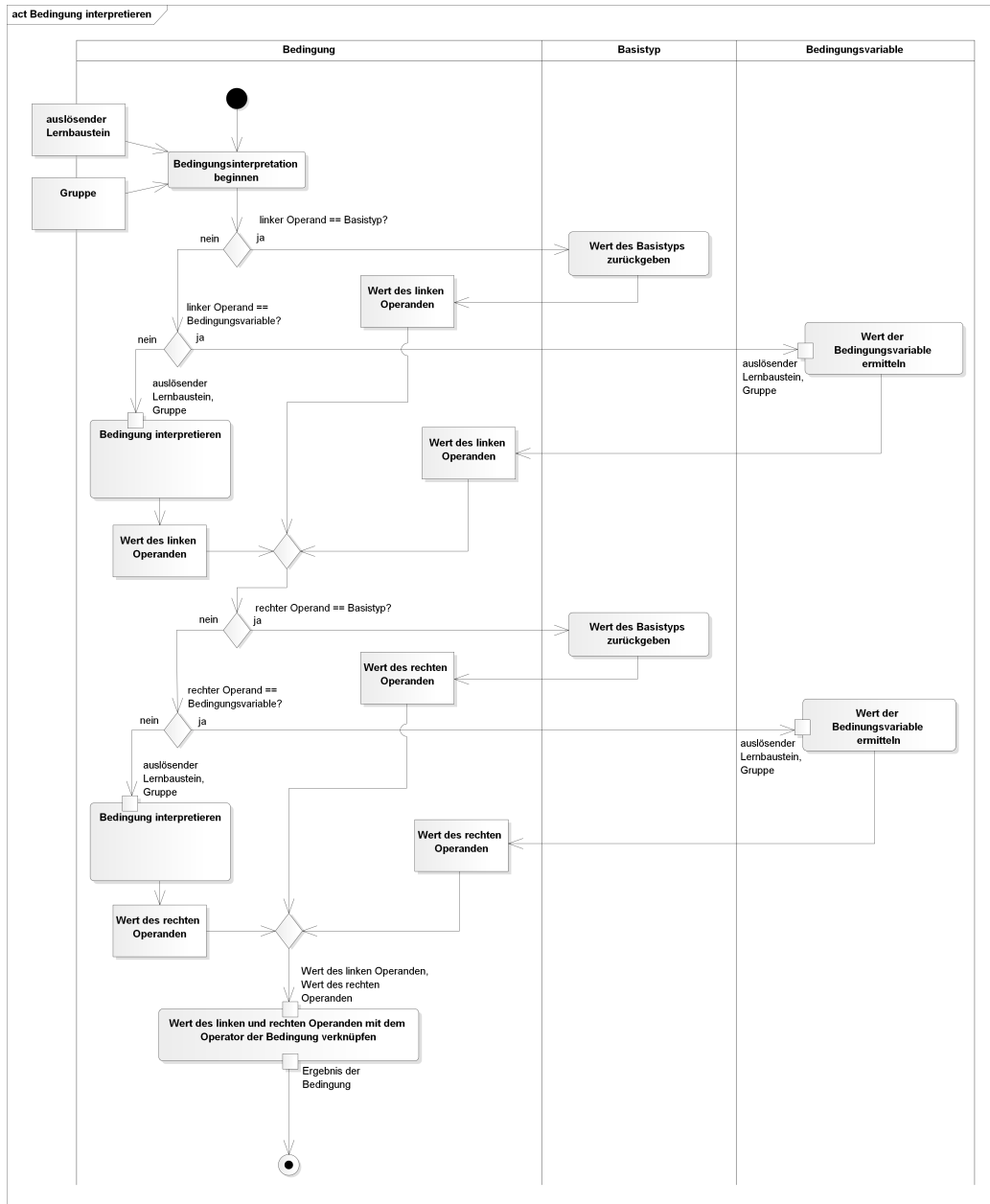


Abbildung 77: Aktivitätsdiagramm - Bedingung interpretieren

Für den rechten Operanden der Bedingung wird die gleiche Prozedur angewandt. Nachdem sowohl der Wert des linken als auch des rechten Operanden ermittelt wurde, werden die beiden Werte mit dem Operator der Bedingung verknüpft. Das Resultat wird dann als Evaluationsergebnis in die Aktivität *Regel evaluieren* zurückgegeben, wo alle Evaluationsergebnisse von Bedingungen miteinander logisch verundet werden. Der Code-Auszug 16 zeigt



beispielhaft die Implementierung der Java Interface Methode *interpretiere(...)*<sup>5</sup> der konkreten Bedingungsklasse *UnGleich*.

Code 16: Implementierte Methode zum Interpretieren eines Ausdrucks aus der LMS Bedingungsklasse *UnGleich*

```

1  @Override
2  public Object interpretiere(Lernbaustein ausloesenderLernbaustein,
3      Gruppe gruppe){
4      Object result;
5      if (linkerOperand instanceof BasisTyp<?>){
6          BasisTyp<?> typeLinks = (BasisTyp<?>) this.linkerOperand;
7          if (this.rechterOperand instanceof BasisTyp<?>){
8              BasisTyp<?> typeRechts = (BasisTyp<?>) this.rechterOperand;
9              result = !typeLinks.getValue().equals(typeRechts.getValue());
10             }
11         else{
12             result = !typeLinks.getValue().equals(this.rechterOperand.
13                 interpretiere(ausloesenderLernbaustein, gruppe));
14         }
15     }
16     else{
17         if (this.rechterOperand instanceof BasisTyp<?>){
18             BasisTyp<?> typeRechts = (BasisTyp<?>) this.rechterOperand;
19             result = !this.linkerOperand.interpretiere(
20                 ausloesenderLernbaustein, gruppe).equals(typeRechts.getValue()
21                 );
22         }
23         else{
24             result = !this.linkerOperand.interpretiere(
25                 ausloesenderLernbaustein, gruppe).equals(this.rechterOperand.
26                 interpretiere(ausloesenderLernbaustein, gruppe));
27         }
28     }
29     return result;
30 }

```

Ist demnach ein Evaluationsergebnis einer interpretierten Bedingungen *false*, wird das gesamte Evaluationsergebnis *false* und die Regel wird die ihr zugeordneten Konsequenzen nicht ausführen (siehe Aktivität *Regeln für Lernbaustein überprüfen*). Ist das gesamte Evaluationsergebnis jedoch positiv, werden alle der Regel zugeordneten Konsequenzen ausgeführt, also je nach Regeltyp entweder *GameDesignElemente* vergeben oder *Feedback* angezeigt und die Aktualisierung der Datenbank angestoßen.

<sup>5</sup> Diese Methode stammt aus dem Interface *IAusdruck* und nutzt eine Wildcard für den Rückgabotyp der Methode, da es sich dabei entweder um ein Objekt vom Typ Boolean, String, Integer oder Lernbaustein (im Fall von Aufgabentyp-Vergleichen) handeln kann.

Wie bereits erwähnt werden Regeln aufgrund von Benutzeraktionen auf Lernbausteinen getriggert. Regeln können jedoch ebenso per Definition mit Lernelementen assoziiert sein, also auch an Gruppen hängen. Das Triggern solcher Regeln, die mit Gruppen assoziiert sind, findet automatisch statt und zwar nach jeder Überprüfung der Regeln eines triggerenden Lernbausteins. Hierfür wird die übergeordnete Gruppe des triggerenden Lernbausteins ermittelt. Sofern diese vorhanden ist, werden die Regeln mit der gleichen Benutzeraktion (mit dem Namenszusatz 'Gruppe'), wie die des triggerenden Lernbausteins überprüft (siehe Abbildung 75), welche mit dieser Gruppe assoziiert sind. Diese gesonderte Überprüfung für Gruppenregeln ist der Tatsache geschuldet, dass Benutzeraktionen auf Lernbausteinen den Zustand von Variablen verändern können (z. B. *AnzKorrektBeantworteteAufgaben*), die sich nicht auf Lernbausteine sondern auf Gruppen beziehen.

#### 7.3.5 Datenbank

Für den Prototyp des Emendo LMS wurde eine lokale und relationale *SQLite*<sup>6</sup> Datenbank verwendet. Diese wird über eine Java Database Connectivity (JDBC)-Verbindung angesprochen, welche als Singleton (Gamma et al., 1995, S. 144ff.) umgesetzt wurde. Die Verbindung zur Datenbank wurde als Teil einer Datenzugriffsschicht gekapselt. Der Zugriff auf eine andere, nicht lokale Datenbank ist demnach problemlos möglich, ohne die Funktionalität der bereits angesprochenen Persistence Repositories anzupassen.

Um die fachlichen Objekte des LMS und ihre dynamischen Daten in die Datenbank zu serialisieren, wurde auf das *OrmLite*-Framework<sup>7</sup> zurückgegriffen. Das Framework erlaubt ein objektrelationales Mapping zwischen Java-Objekten und SQL-Datenbanken. Dafür ist es notwendig die zu persistierenden Java Objekte mit Annotationen zu erweitern, um beispielsweise ein Attribut einer Klasse so zu annotieren, dass es in einer Datenbanktabelle als Feld auftritt. Das Ergebnis dieser Annotationen und des zuvor erfolgten logischen Datenbankdesigns in Form eines Entity-Relationship (ER)-Modells ist durch das in Abbildung 78 dargestellte physische SQL-Datenbankmodell dokumentiert.

<sup>6</sup> Siehe <https://www.sqlite.org/>, abgerufen am: 12.09.2017

<sup>7</sup> Siehe <http://ormlite.com/>, abgerufen am: 12.09.2017

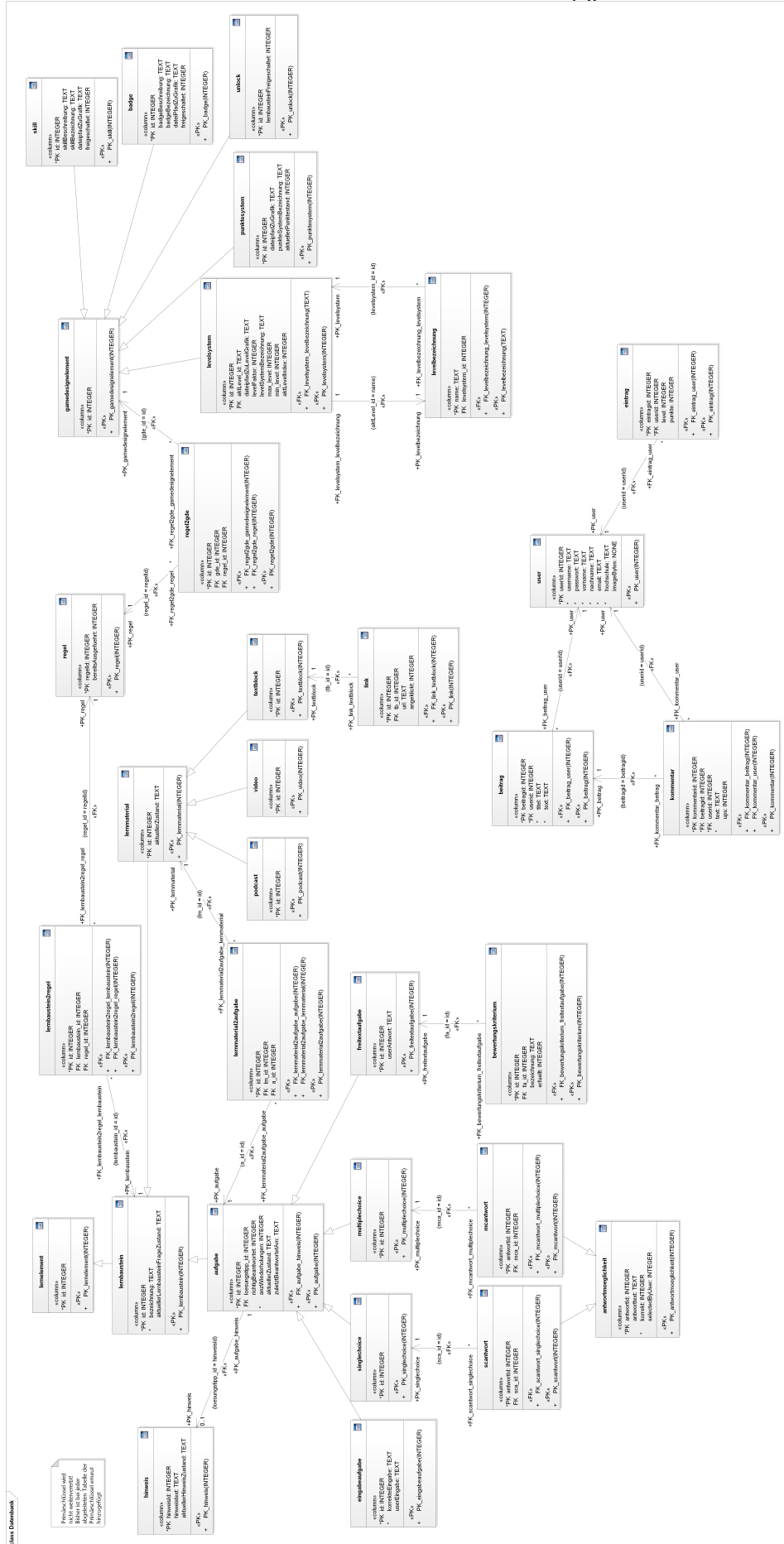


Abbildung 78: SQL-Datenbankmodell von Emendo LMS

Ebenso stellt *OrmLite* eine standardisierte Methode zur Abfrage und Manipulation von Daten über das Muster Data Access Object (DAO) (Oracle Corporation, 2011) zur Verfügung. Dabei ist jeder in SQL persistierten Fachklasse eine DAO-Klasse zugeordnet, die die Formulierung von Queries entweder über die API von *OrmLite* oder direkt in SQL erlauben. Die entsprechenden DAOs sind Teil der Persistence Repositories und bilden damit eine gekapselte Datenzugriffsschicht auf die Datenhaltung. Der Code Auszug 17 zeigt beispielhaft eine Methode des Persistence Repositories von *EingabeAufgaben*. Darin ist die Verwendung der *OrmLite* API (z. B. *QueryBuilder*, *PreparedQuery*) in Verbindung mit einer DAO (*eaDAO*) zu erkennen, die benutzt wird, um eine Query zu erstellen, die vom Framework intern in eine SQL-Query umgewandelt wird. Nach dem Absetzen der Query durch die DAO wird das Ergebnis der Abfrage an die aufrufende Instanz zurückgegeben, sofern kein Fehlerfall eintritt.

Code 17: Beispielhafte Methode des Persistence Repositories von *EingabeAufgaben*

```
1 public int getAlleKorrektBeantwortetenEingabeAufgaben(){
2     int res = 0;
3     try {
4         QueryBuilder <EingabeAufgabe, Integer> queryBuilder = eaDAO.
5             queryBuilder();
6         queryBuilder.where().eq("richtigBeantwortet", true);
7         PreparedQuery <EingabeAufgabe> preparedQuery = queryBuilder.prepare
8             ();
9         res = eaDAO.query(preparedQuery).size();
10    }
11    catch (SQLException e) {
12        logger.error("Error in getAlleKorrektBeantwortetenEingabeAufgaben()")
13        );
14        e.printStackTrace();
15    }
16    return res;
17 }
```

## 7.4 GRAFISCHE BENUTZERSCHNITTSTELLE

Im Folgenden soll kurz auf die GUI des Emendo LMS eingegangen werden, unter anderem um aufzuzeigen, wie die Umsetzung der restlichen konzeptuellen Anforderungen aus Tabelle 15 in das Emendo LMS eingeflossen sind. Die GUI des Emendo LMS wurde mit *WebLaF*<sup>8</sup> realisiert, ein Java Framework für Swing Applikationen.

<sup>8</sup> Siehe <http://weblookandfeel.com/>, abgerufen am: 17.09.2017

### 7.4.1 Hauptpanels

Wie bereits dargestellt, besteht die GUI des Emendo LMS aus Hauptpanels, die durch die Komponente *MainPanelControllerManager* verwaltet werden. Die Hauptpanels, sowie alle weiteren Bildschirmseiten, wurden zunächst als Papierprototypen (Heinsen u. Vogt, 2003, S. 261f.) erstellt. Abbildung 79 zeigt beispielhaft den Papierprototypen der *Home*-Bildschirmseite.

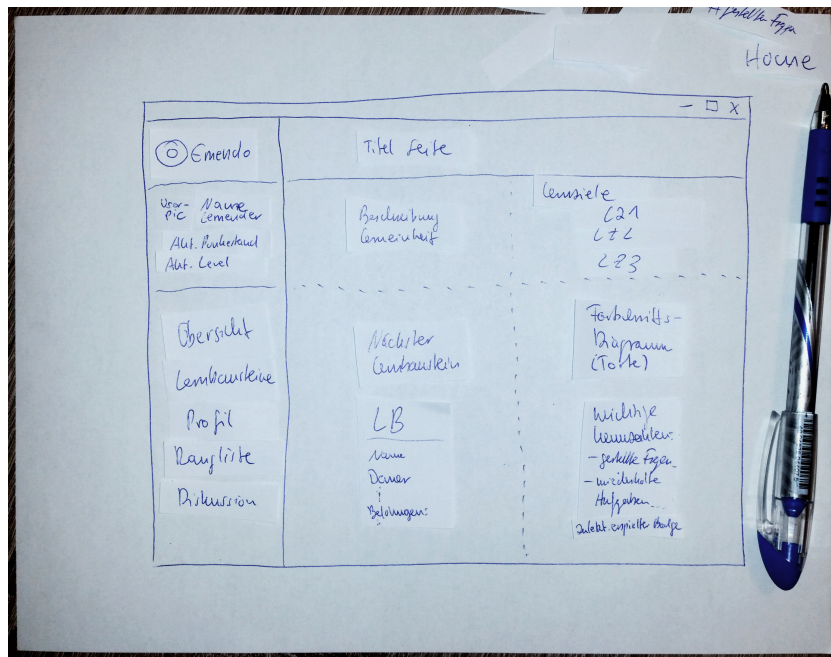


Abbildung 79: Papierprototyp der Emendo LMS Bildschirmseite Home

Die folgenden Hauptpanels, im Sinne von Bildschirmseiten, sind im Emendo LMS verfügbar und werden kurz beschrieben (Screenshots zu allen relevanten Bildschirmseiten befinden sich im Anhang A.6):

- **Home:** Diese Bildschirmseite stellt allgemeine Informationen zur Lerneinheit bereit, gibt die Möglichkeit für einen Schnelleinstieg in die Bearbeitung des nächsten freigeschalteten Lernbausteins und gibt eine Übersicht über den aktuellen Bearbeitungsfortschritt zur Lerneinheit.
- **Lernbausteine:** Diese Seite gibt eine Übersicht über alle Lernbausteine (inkl. Detailinformationen, wie Belohnungen zu jedem Lernbaustein gemäß B\_Z1 und Anzahl Fehlversuche beim Lösen von Aufgaben nach A\_A7), die in der Lerneinheit verfügbar sind (siehe beispielhaft Abbildung 80). Sie zeigt zudem an, ob einzelne Lernbausteine freigeschaltet sind (CU\_A2) und visualisiert Lernpfade durch ein *Hovering* über einzelne Lernbausteine. Durch einen Klick auf einen freigeschalteten Lern-

baustein, kann die Bearbeitung dessen in einer neuen Bildschirmseite gestartet werden.

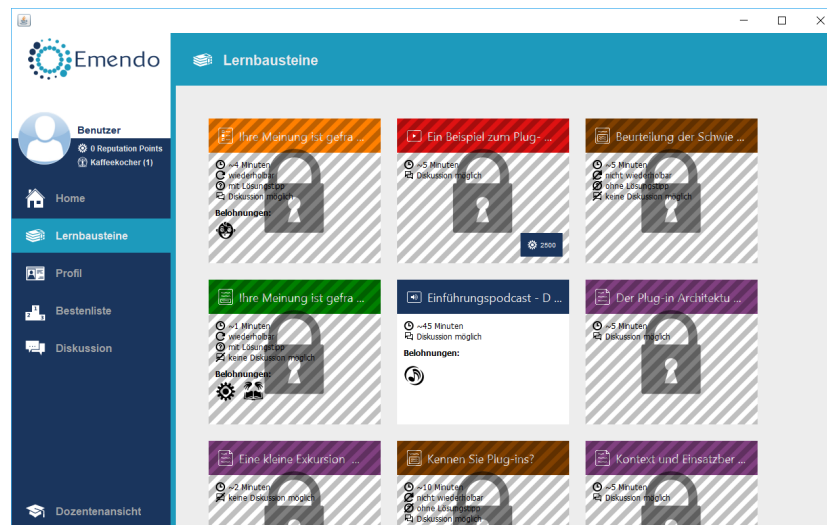


Abbildung 80: Screenshot der Lernbausteinübersicht im Emendo LMS

- **Profil:** Die Profilsseite (siehe Abbildung 81) gibt Lernenden Auskunft über den Lernfortschritt innerhalb einer Lerneinheit (PR\_A2). Sie präsentiert Informationen zu dem angemeldeten Benutzer (PR\_A1), erhaltenen *GameDesignElementen* (B\_A3 mit Einschränkung der Verteilung in sozialen Netzwerken, PR\_Z1) sowie visualisiert den Lernbausteinfortschritt. Die Daten des Profils werden beim Aufruf aktualisiert (PR\_R1).

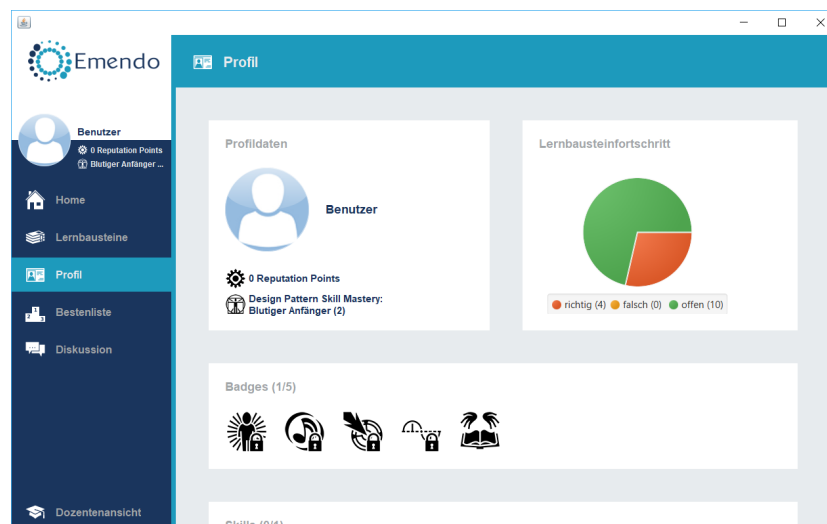


Abbildung 81: Screenshot der Profilsseite im Emendo LMS

- **Bestenliste:** Die Bestenliste erlaubt einen Vergleich mit anderen Lernenden innerhalb der Lerneinheit (LB\_A2, LB\_Z1). Sie bietet zudem Sor-

tierungsmöglichkeiten, anhand welcher Metrik dieser Vergleich stattfinden soll (LB\_A1, B\_Z2). Der Vergleich beschränkt sich auf die an der Lerneinheit teilnehmenden Lernenden (LB\_A3).

- **Diskussion:** Im Diskussionsbereich werden Fragen, die zu Lernbausteinen während deren Bearbeitung gestellt werden (D\_Z1), strukturiert in Form von einzelnen Diskussionen zusammengetragen (D\_A3). Die Anzahl an Diskussionen ist ausgewiesen (D\_A1). Diskussionen können von Lernenden kommentiert werden (D\_A4). Zu jeder Diskussion ist die Anzahl an vorhandenen Kommentaren dokumentiert (D\_A2). Kommentare können von Lernenden als hilfreich markiert werden (D\_A5). Auf die Umsetzung von einer Suchfunktionalität<sup>9</sup> innerhalb von Kommentaren wurde an dieser Stelle verzichtet (D\_A6).

#### 7.4.2 *Bearbeitung von Lernbausteinen*

##### 7.4.2.1 *Widgets*

Eine Bearbeitung eines Lernbausteins kann entweder über die *Home-* oder *Lernbausteine*-Bildschirmseite begonnen werden. Je nach Lernbausteintyp enthält die Bildschirmseite verschiedene Inhalte, die durch Widgets repräsentiert sind und einem konsistenten Aufbau folgen. Durch die Zuordnung von einem Lernbausteintyp aus der Emendo DSML zu einem entsprechenden Pendant im Emendo LMS, wobei Letzteres als MVC-Triade umgesetzt ist, erlaubt diese Strukturierung, dass mehrere Views in dieser Triade enthalten sein können. Daraus folgt, dass es beispielsweise mehrere Widgets geben kann, die ein Lernbaustein-Konzept abbilden können.

##### 7.4.2.2 *Benachrichtigungen*

Während der Bearbeitung von Lernbausteinen können abhängig von den assoziierten Regeln die folgenden Benachrichtigungen auftreten: Erhalt von Badges (B\_A4) oder Points (P\_A2), die Veränderung von Level- oder Skillzuständen (L\_A3 bzw. S\_A2), das Abschließen von Quests (Q\_A3) und das Freischalten von Unlocks (CU\_A1). Die Abbildung 82 zeigt beispielhaft den Benachrichtigungs-Popup für den Erhalt eines Badges, der als Belohnungskonsequenz einer Regel vergeben wird. Benachrichtigungen als Folge von Belohnungskonsequenzen besitzen einen generischen Aufbau und bestehen aus einem

<sup>9</sup> Eine Möglichkeit diese Funktionalität zu implementieren, stellt das Java-basierte Framework Apache Lucene (siehe <https://lucene.apache.org/>, abgerufen am: 19.09.2017) dar.

Aktionsnamen, einem Icon sofern vorhanden und einer entsprechenden Erklärung.



Abbildung 82: Benachrichtigung für den Erhalt eines Badges

#### 7.4.2.3 Vorschläge für bearbeitbaren Lernbaustein

Ist eine Bearbeitung eines Lernbausteins vollständig abgeschlossen, schlägt das Emendo LMS Lernenden einen nächsten freigeschalteten und damit bearbeitbaren Lernbaustein vor (Q\_Z2). Hierfür nutzt das LMS den in Pseudocode formulierten Algorithmus, dargestellt in Code Auszug 18.

Code 18: An C angelehnter Pseudocode-Algorithmus zum Ermitteln des nächsten bearbeitbaren Lernbausteins

```

1
2 // Deklarationen
3 LernbausteinPfad aktLernbausteinPfad;
4 LernbausteinPfade lernbausteinPfade;
5 LernbausteinPfade ueberpruefteLernbausteinPfade;
6
7 Lernbaustein sucheNaechstenLernbaustein(Lernbaustein aktLernbaustein) {
8     int lernbausteinIndex;
9
10    lernbausteinPfade = ermittleAlleLernbausteinPfade();
11    aktLernbausteinPfad = lernbausteinPfade.
        ermittleLernbausteinPfadMitKleinstenIndexFuerLernbaustein(
            aktLernbaustein);
12    lernbausteinIndex = aktLernbausteinPfad.ermittleIndexFuerLernbaustein(
        aktLernbaustein);
13
14    return durchsucheAktLernbausteinPfad(lernbausteinIndex);
15 }
16

```



```

17 Lernbaustein durchsucheAktLernbausteinPfad(int lernbausteinIndex) {
18     if(lernbausteinIndex == aktLernbausteinPfad.laenge - 1) {
19         ueberpruefteLernbausteinPfade.fuegePfadHinzu(aktLernbausteinPfad);
20         return durchsucheNaechstenLernbausteinPfad(lernbausteinIndex);
21     }
22     else {
23         if(aktLernbausteinPfad[lernbausteinIndex + 1].istFreigeschaltet()) {
24             return aktLernbausteinPfad[lernbausteinIndex + 1];
25         }
26         else {
27             return durchsucheAktLernbausteinPfad(lernbausteinIndex + 1);
28         }
29     }
30 }
31
32 Lernbaustein durchsucheNaechstenLernbausteinPfad(int lernbausteinIndex)
33 {
34     LernbausteinPfad naechsterLernbausteinPfad;
35
36     while(lernbausteinIndex > 0) {
37         if(aktLernbausteinPfad[lernbausteinIndex - 1].istFreigeschaltet()) {
38             return aktLernbausteinPfad[lernbausteinIndex - 1];
39         }
40         else {
41             naechsterLernbausteinPfad = sucheNaechstenLernbausteinPfad(
42                 lernbausteinIndex);
43             if(naechsterLernbausteinPfad == null) {
44                 lernbausteinIndex--;
45             }
46             else {
47                 aktLernbausteinPfad = naechsterLernbausteinPfad;
48                 return durchsucheAktLernbausteinPfad(lernbausteinIndex - 1);
49             }
50         }
51     }
52     return null;
53 }
54
55 LernbausteinPfad sucheNaechstenLernbausteinPfad(int lernbausteinIndex) {
56     LernbausteinPfade nichtUeberpruefteLernbausteinPfade;
57     LernbausteinPfad teilPfad;
58
59     nichtUeberpruefteLernbausteinPfade = lernbausteinPfade.
60         ermittleNochNichtUeberpruefteLernbausteinPfade(
61             ueberpruefteLernbausteinPfade);
62     teilPfad = aktLernbausteinPfad.ermittleTeilPfadBisIndex(
63         lernbausteinIndex);
64
65     return nichtUeberpruefteLernbausteinPfade.
66         sucheErstenLernbausteinPfadMitTeilPfadAlsAnfangssequenz(teilPfad);
67 }

```

Der Algorithmus ist als konkrete Strategie gemäß dem Strategie-Muster (Gamma et al., 1995, S. 349ff.) implementiert. Das Emendo LMS kann demnach um weitere Algorithmen zur Findung eines nächsten bearbeitbaren Lernbausteins ergänzt werden. Beispielsweise könnten Künstliche Intelligenz (KI)-Techniken dafür eingesetzt werden, konkrete Strategien zu entwickeln, die den nächsten bearbeitbaren Lernbaustein auf Basis der Performanz bei vorher bearbeiteten Lernbausteinen vorschlagen.

Derartige Strategien würden die Adaptivität (Rey, 2009, S. 179) des Emendo LMS steigern. Darüber hinaus kann ein weiterer Beitrag zur Adaptivität der dynamische Austausch von Controllern in MVC-Triaden sein. Dieser erfolgt, um die Schwierigkeit einer Aufgabe adaptiv anzupassen, indem beispielsweise bei einer Single Choice-Aufgabe statt der vorgesehenen sechs Antwortmöglichkeiten nur drei davon Lernenden angezeigt werden, was zu einer Verringerung der Schwierigkeit führt. Umgekehrt ist diese Anpassung ebenso denkbar, denn:

„The challenge is not getting people to want to learn and grow – they already do – it is helping them find the path to success“ (Burke, 2014a, S. 69).

#### 7.4.2.4 *Einschränkungen*

Insgesamt wurden drei konzeptuelle Anforderungen nicht umgesetzt. Konkret handelt es sich bei zwei davon um das Verbreiten von Informationen zum Erhalt von Badges (B\_A3) und Points (P\_A3) in sozialen Netzwerken. Auf eine Umsetzung wurde in diesem Stadium des Prototyps verzichtet, da dies nicht zu der Kernfunktionalität eines Prototypen des Emendo LMS zählt. Darüber hinaus ist es in dieser Version nicht möglich Aufgaben mit Lesezeichen zu versehen (A\_A6). Dies ist der Tatsache geschuldet, dass Emendo auf Ebenen von Lerneinheiten unterscheidet und nicht auf den Ebenen von ganzen Semesterkursen, die aus mehreren Lerneinheiten bestehen können. Daraus folgt, dass die Anzahl an Lernbausteinen wesentlich kleiner und damit für Lernende überschaubarer ist, was die Lesezeichen-Funktion nicht zwingend notwendig macht.

#### 7.4.3 *Überwachung des Lernfortschritts und Lehrendenfeedbacks*

Aus Sicht der Lehrenden kann der Lernfortschritt von Lernenden überwacht werden. Das Emendo LMS stellt hierzu die folgenden beiden Bildschirmseiten

bereit, die in dieser Version aus Gründen der Einfachheit mit in die Anwendung des LMS integriert wurden, in einer späteren Version jedoch in einer separaten Instanz lauffähig sein sollen (Screenshots befinden sich ebenso in Anhang A.6):

- **Studentenfortschritt:** Diese Bildschirmseite gibt für Lehrende eine quantitative Übersicht, welche Lernbausteine der Lerneinheit wie häufig bereits erfolgreich bearbeitet wurden.
- **Manuelle Bewertung:** Lehrende haben hier die Möglichkeit, nicht automatisch bewertbare Lösungsformate, wie die der Freitextaufgaben, manuell anhand von Bewertungskriterien zu bewerten. Hierfür können die Antworten von Lernenden einzeln eingesehen und anhand der Kriterien eine Bewertung abgegeben werden. Außerdem können in den Lernendenantworten enthaltene Textbausteine direkt in Präsentationsfolien übernommen werden, die direkt aus dem Emendo LMS in \*.pptx exportiert werden können – ein im JiTT (Novak et al., 1999) übliches Vorgehen.

## 7.5 MOBILE EMENDO LMS

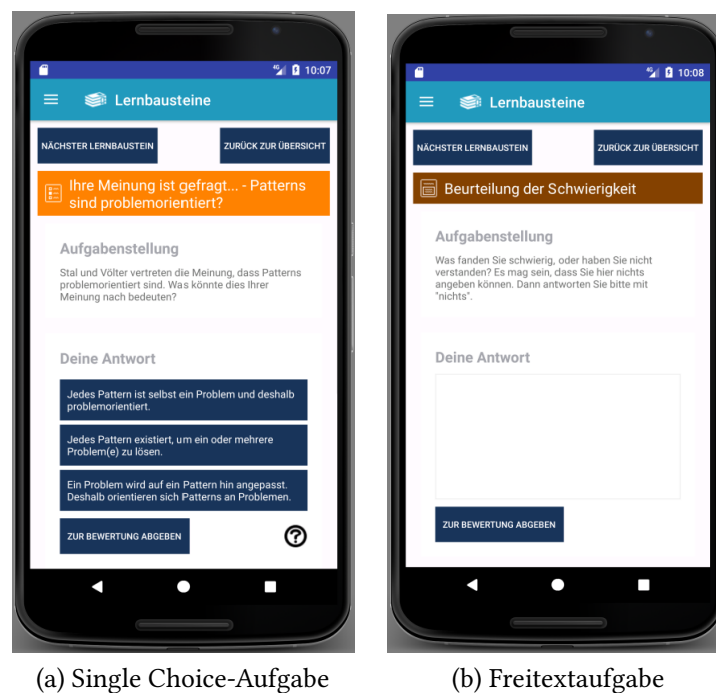
Zu Demonstrationszwecken der Portabilität des Emendo LMS wurde eine Bachelorarbeit im Umfeld dieser Arbeit erstellt, die eine Portierung des LMS auf die Android Plattform untersuchen und durchführen sollte (Saliger, 2018). Ziel bei dieser Portierung war neben einer Machbarkeitsuntersuchung ebenso den Style der GUI des LMS weitestgehend beizubehalten.

Die Basis für die Portierung bildete der zuvor beschriebene Einsatz des MVC-Architekturmusters, bei dem in der Theorie auf unterschiedlichen Abstraktionsstufen jeweils das Model im Zuge einer Portierung auf eine andere Zielumgebung unangetastet bleibt und die View sowie Teile des Controllers gemäß der neuen Zielumgebung angepasst werden müssen. Auch wird eine derartige Portabilität dadurch ermöglicht, dass der Output des Emendo Generators Quellcode in der Programmiersprache Java ist, was ebenfalls eine mögliche Programmiersprache der Android-Umgebung darstellt. Damit kann mit ein und demselben Generator bereits jetzt nicht nur eine Client-Anwendung für Desktop-Systeme bedient werden, sondern darüber hinaus auch eine mobile Android Applikation.

Gemäß Saliger (2018) waren für die Portierung konkret die folgenden Anpassungen notwendig:

- Implementieren des *MainFrames* als *Activity* und der einzelnen MVC-Views (Panels) als Fragments.
- Austausch aller *Swing*- und *WebLaF*-Komponenten durch Android-Komponenten.
- Einführen von *FragmentLifecycleCallbacks*, die benötigt werden, weil Android eine Initialisierung der Elemente auf einer View erst zulässt, wenn die View aufgerufen wird.
- Anpassung der Listener-Klassen der Controller, aufgrund von Änderungen der Elementtypen auf den Views.
- Anpassung der Datenbankverbindung und Ersetzung von *OrmLite* durch *OrmLite Android*.

Screenshots<sup>10</sup> der auf Android adaptierten Bildschirmseiten und damit verbundene Funktionalitäten befinden sich in Anhang A.7. Exemplarisch sei an dieser Stelle mit Abbildung 83 ein Beispiel für die Bearbeitung einer Single Choice-Aufgabe sowie einer Freitextaufgabe gezeigt.



(a) Single Choice-Aufgabe

(b) Freitextaufgabe

Abbildung 83: Screenshots des Emendo LMS Mobile – Single Choice-Aufgabe und Freitextaufgabe

<sup>10</sup> Bereitgestellt im Rahmen der Anfertigung der Abschlussarbeit (Saliger, 2018).

## 7.6 ZWISCHENFAZIT

Dieses Kapitel gab aus Software Engineering Sicht Einblicke in das Emendo LMS. Es stellte die Vorüberlegungen des LMS in Form einer Anforderungsanalyse vor und präsentierte die darauf aufbauende Systemarchitektur und dabei die Funktionalität von relevanten Komponenten. Zudem wurde die GUI und dabei relevante Designentscheidungen aus Lernenden- und Lehrendensicht vorgestellt und diskutiert. Ebenso wurde auf die Portabilität des Emendo LMS auf die Android Plattform eingegangen und gezeigt, dass auf Basis der DSML und des Generators mit vergleichsweise wenig Anpassungsaufwand zwei Zielplattformen bedient werden können.

Das darauffolgende Kapitel widmet sich der Evaluation von ausgewählten Systembestandteilen von Emendo.

---

## EVALUATION

---

*„Wer aufhört, besser werden zu wollen,  
hört auf, gut zu sein.“*

*— Marie von Ebner-Eschenbach*

### 8.1 EINLEITUNG

Nach dem DSR sollen entwickelte Artefakte sich evaluativen Methoden unterziehen, um sie mit analytischen Studien, Fallstudien, Experimenten, Feldstudien oder Simulationen zu bewerten (Hevner et al., 2004, S. 80). Die Emendo Werkzeugkette stellt für eine derartige Beurteilung deren Systembestandteile bestehend aus der Emendo DSML, dem Designer, dem Generator und schließlich dem LMS zur Verfügung. Die in diesem Kapitel vorgestellte Evaluation umfasst zwei Arten: Zum einen wurde eine Feldstudie mit potentiell zukünftigen Nutzern von Emendo durchgeführt und zum anderen erfolgt eine funktionale Evaluation anhand eines Abgleichs von erhobenen Anforderungen zu umgesetzten Artefakten. Der Fokus beider Evaluationen soll auf der Benutzergruppe der Lehrenden liegen. Diese Benutzergruppe nimmt die Konzeption von spielifizierten Lehr-Lernelementen in formalen Lernkontexten mit der DSML bzw. dem Designer vor und entscheidet über die didaktisch sinnvolle Platzierung dieser Einheiten im Rahmen von Lehrveranstaltungen. Zudem erscheint aus den Aussagen dieser Personengruppe die Beurteilung der Lösung von bereits auf der Literatur basierenden Forschungsdesideraten besser geeignet, als die Personengruppe der Lernenden. Die Evaluation mit Letzteren würde aufgrund des prototypischen Charakters des Emendo LMS bedingen, dass deutlicher Mehraufwand für das Erreichen einer Produktreife des LMS investiert werden müsste, was den Anspruch an einen Proof-of-Concept weit übersteigen würde. Demnach wird in diese Evaluation der Generator bzw. das LMS nicht einbezogen. Darüber hinaus wird beispielhaft am Konzept des spielifi-

zierten JiTT gezeigt, wie Emendo eingesetzt werden kann, um eine Lerneinheit für ein Entwurfsmuster aus dem Software Engineering zu designen.

## 8.2 EVALUATION DER DSML UND DES DESIGNERS

Das folgende Kapitel beschreibt die eingangs erwähnte Feldstudie zur Evaluation der Emendo DSML und des Emendo Designers.

### 8.2.1 *Ziele und Vorüberlegungen*

Die Überlegungen, die dem Konzept von Emendo zugrunde liegen, entstammen nach DSR sowohl aus der Wissensbasis als auch aus der Realweltumgebung. Es erfolgte jedoch bisher keine Beurteilung seitens potentieller Anwender. Obwohl die DSML und der Designer zwei wesentliche Artefakte von Emendo darstellen, kann deren Beurteilung nicht isoliert von den restlichen Artefakten stattfinden. Wäre dies der Fall, würde die Einschätzung der beiden Artefakte lediglich auf einem begrenzten Kenntnisstand der Probanden erfolgen und der potentielle Nutzen von Emendo nicht vollständig repräsentiert. Interessant erscheint also eine Beurteilung des Gesamtkonzepts von Emendo durch Lehrende und dessen Möglichkeiten der Integration in den Lehralltag an Hochschulen sowie damit verbundene Erwartungen in Bezug auf Vor- und Nachteile eines solchen Einsatzes. Es soll also eine Aussage darüber gemacht werden können, ob Lehrende einen Vorteil durch einen Einsatz von Emendo (im Vergleich zu der jeweils aktuell vorherrschenden Lehrsituation) sehen und wie bzw. unter welchen Bedingungen sich dieser Vorteil in den praktischen Einsatz als didaktischer Bestandteil von Lehrveranstaltungen transferieren lässt. Hierbei genannte Aspekte können nicht nur Erkenntnisse darüber liefern, wie Emendo eingeschätzt wird, sondern auch Dinge aufdecken, die sinnvolle Erweiterungen für Emendo darstellen können. Neben der Erwartungshaltung gegenüber Emendo und damit verbundenen theoretischen Überlegungen für einen Einsatz erscheint es darüber hinaus interessant zu erfahren, wie Lehrende die Erstellung von spielifizierten Lehr-Lernarrangements beurteilen. Zur Erreichung dieses Ziels bedarf es einer praktischen Erprobung der DSML und den Umgang damit im Designer. Hierbei wäre zu erfassen, ob und wie selbsterklärend sich spielifizierte Lehr-Lernarrangements durch Lehrende mit dem Sprachumfang der DSML und den funktionalen Möglichkeiten des Designers umsetzen lassen. Aus diesen Über-

legungen ergeben sich die folgenden Forschungsfragen, die diese Evaluation<sup>1</sup> leiten sollen:

- **FFE\_1:** Wie schätzen Lehrende die Einsetzbarkeit von Emendo in Lehrveranstaltungen ein?
- **FFE\_2:** Unter welchen Rahmenbedingungen erfolgt dieser Einsatz?
- **FFE\_3:** Wie wird der Sprachumfang der Emendo DSML von Lehrenden beurteilt?
- **FFE\_4:** Wie werden die Selbsterklärungsfähigkeit der Bestandteile und deren Verwendung der Emendo DSML von Lehrenden eingeschätzt?
- **FFE\_5:** Wie schätzen Lehrende die Funktionalität des Emendo Designers zum Modellieren mit der Emendo DSML ein?

### 8.2.2 Methodische Vorüberlegungen

Für die Beantwortung der genannten Forschungsfragen bedarf es eines fundierten methodischen Vorgehens, welches in der Lage ist, Theorien bzw. Meinungen von Probanden vor und während der Interaktion mit den Artefakten systematisch zu erfassen und miteinander in Bezug zu setzen. Konkret zielt diese Evaluation also auf die Erfassung *subjektiver Theorien* ab. Darunter verstehen Christmann et al.

„[...] Kognitionen des Alltagsmenschen [...], die sich inhaltlich auf seine Selbst- und Weltsicht beziehen und eine vergleichbare Struktur aufweisen, wie objektive Theorien für den/die Wissenschaftler/in, nämlich die der Erklärung, Prognose und Technologie (Christmann et al., 1999, S. 138f.).

Der Begriff der Theorie in diesem Kontext wird von Christmann et al. und Scheele u. Groeben interpretiert als komplexe Aggregate von menschlichen Kognitionen mit implizierter Argumentationsstruktur, welche im Dialog-Konsens aktualisier- und rekonstruierbar ist (Christmann et al., 1999, S. 138; Scheele u. Groeben, 1988, S. 15).

Der Erfassung von subjektiven Theorien hat sich das *Forschungsprogramm subjektive Theorien* (Groeben u. Scheele, 2010) gewidmet und die *Dialog-Konsens-Methodik* hervorgebracht. Die Dialog-Konsens-Methodik sieht üblicherweise

<sup>1</sup> Sowohl das Vorgehen als auch relevante Ergebnisse der Evaluation sind in publiziert in Bartel et al. (2018).



zwei voneinander getrennte Phasen vor, nämlich eine Erhebungsphase für Theorieinhalte und eine Rekonstruktionsphase in der die Strukturierung der Theorie erfolgt (Christmann et al., 1999, S. 140; Groeben, 1992). Für die erste Phase werden typischerweise etablierte Erhebungsverfahren verwendet, wie (halbstandardisierte) Interviews (Groeben, 1992). Jedoch sind auch andere Erhebungsverfahren möglich, wie beispielsweise Fragebögen oder Fokusgruppen (Heinsen u. Vogt, 2003, S. 107ff.). Derartige Erhebungsverfahren fragen eine bereits strukturierte Realität ab und es kann keine Validierung dahingehend erfolgen, dass Probanden diese Vorstrukturierung teilen (Christmann et al., 1999, S. 139). Deshalb sind sie für die Beantwortung der Forschungsfragen weniger geeignet. Es folgt in der zweiten Phase eine Strukturierung der Inhalte durch die Struktur-Lege-Technik (SLT)<sup>2</sup> (Scheele u. Groeben, 1988). Die SLT hat zum Ziel, im Dialog zwischen Forscher und Proband Einigungen in Bezug auf „[...] subjektive Reflexion[en], Überzeugung[en], Erklärung[en] des Erkenntnis-Objekts über einen bestimmten Inhaltsbereich möglichst vollständig, umfassend und unverzerrt [...]“ (Christmann et al., 1999, S. 139) zu erzielen, diese auf Karteikärtchen zu visualisieren und mit anderen Einigungen in Bezug zu setzen (Scheele u. Groeben, 1984). Nachdem die zweite Phase abgeschlossen ist, sind Daten erhoben, deren Validität sich ausschließlich auf eine Testperson bezieht. Um eine Vergleichbarkeit zwischen den Daten herzustellen, muss auf eine Vergleichbarkeit der Probanden in Bezug auf persönliche Kriterien geachtet werden (Christmann et al., 1999, S. 147). Es folgt eine systematische Auswertung der Daten nach etablierten Verfahren, die entweder auf eine Struktur- oder Inhaltsebene (siehe hierzu z. B. Mayring (2010)) abzielen und quantitativ oder qualitativ formalisieren (Christmann et al., 1999, S. 147f.).

Die aufgezeigte Eignung dieses Vorgehens zur Beantwortung der genannten Forschungsfragen wird durch einschlägige Literatur zudem unterstützt und der Empfehlung von Heinsen u. Vogt gefolgt. Sie empfehlen die SLT für die Erfassung der Nutzermeinung bzw. Nutzersicht auf Software mit dem Ziel, mit vornehmlich Experten Handlungsstrukturen und Nutzermodelle in Bezug auf Software zu erfassen (Heinsen u. Vogt, 2003, S. 107, 109).

### 8.2.3 Durchführung

Im Folgenden soll das Sample sowie das Vorgehen bei der Durchführung der Evaluation beschrieben werden.

<sup>2</sup> Gemeint ist die Heidelberger SLT (Scheele u. Groeben, 1984). Da weitere Verfahren mit vergleichbaren Bezeichnungen existent sind, bedarf es einer Präzisierung.

### 8.2.3.1 *Sample*

Für die Durchführung der Evaluation wurde eine Samplegröße von 5-10 Personen angestrebt. Die Untergrenze ergibt sich aus der Annahme von Nielsen u. Landauer (1993), welche herausfanden, dass bereits 5 Experten ausreichend sind um 85% aller Usability-Probleme in einer Software zu finden. Da es sich bei der durchgeführten Evaluation nicht ausschließlich um eine Usability-Evaluation handelt, vielmehr auch lerntheoretische und motivationale Aspekte (Konzeption von durch Anreize gestützte Lerneinheiten) eine wichtige Rolle spielen, wurde der Sampleumfang entsprechend erweitert. Die Obergrenze von 10 Personen ergab sich zum einen durch eine Sättigungserwartung bzgl. neuer Erkenntnisse, welche nach Nielsen u. Landauer bereits bei 5 Personen eintritt, und zum anderen bedingt durch den Vorzug von einer tieferen vor einer quantitativ höheren aber damit oberflächlicheren Analysemöglichkeit der Ergebnisse.

Die SLT setzt voraus, dass bestimmte Fähigkeiten motivationaler und kognitiver Art bei Probanden vorliegen müssen (Groeben u. Scheele, 2010; Christmann et al., 1999). Diese Anforderung, gepaart mit der Empfehlung von Heinssen u. Vogt (2003, S. 109), das Sample der SLT aus Experten zusammenzustellen, wurde entsprechend umgesetzt. Als Experten werden nach Kruse (2014, S. 176) Personen bezeichnet, welche über Sonderwissen verfügen. Dabei handelt es sich entweder um Kontextwissen, das Wissensbestände auf ein fachliches und ggf. breites Feld integriert, oder Betriebswissen, das vorliegt, wenn Personen ein selbstreflexives Wissen bezogen auf spezifische Handlungsfelder und -prozesse besitzen (Kruse, 2014, S. 176). Im Kontext der Evaluation sind demnach Experten gleichzusetzen mit Lehrenden, die über Sonderwissen, abhängig von ihrem Lehrgebiet, verfügen.

Für die Evaluation konnten n=8 (3 weiblich, 5 männlich) Experten gewonnen werden. Um eine maximale Kontrastierung und damit eine breite Varianz innerhalb der Experten zu erzielen, wurden Kriterien berücksichtigt, die personenbezogen erhoben wurden. Zu den Kriterien zählten (bildungs-) biographische Angaben, wie Alter, Lehrerfahrung, höchster akademischer Grad, der Fachbereich des höchsten akademischen Grads sowie die dienstliche Stellung. Von besonderer Bedeutung waren Kriterien in Bezug auf die Vorerfahrung mit Gamification und den Einsatz von LMS in der Lehre, da sie besonderes Erkenntnispotenzial, abhängig von den Kriterienausprägungen, für die Beantwortung der Forschungsfragen erhoffen lassen. Diese Kriterien-basierte Auswahl von Probanden kann ebenso als Stichprobenplan (Schreier, 2010, S. 245f.) bezeichnet werden, was für die Evaluation eine Heterogenität der Probanden sicher-

stellt und für (Frank, 2013, S. 155) ein Qualitätsmerkmal zur Evaluation von domänenspezifischen Modellierungssprachen darstellt.

Tabelle 19 zeigt eine Übersicht über die Probanden und deren Expertenprofile. Aus Gründen der Anonymität tragen die einzelnen Probanden alphabetische Bezeichnungen (A-H). Für die Erfassung der Einsatzhäufigkeit von Gamification und LMS wurden jeweils 7-stufige Likert-Skalen verwendet, die aus Gründen der Verständlichkeit verbalen Codes zugeordnet wurden (1-sehr selten bis 7-sehr häufig).

### 8.2.3.2 *Vorgehen bei der Evaluation*

Evaluationen wurden im Zeitraum von November 2017 bis Januar 2018 durchgeführt. Die Interview-durchführende Person war der Autor dieser Arbeit. Die Evaluationen wurden in physischer Anwesenheit von Probanden und Forscher vollzogen, mit Ausnahme von einer Evaluation, die aus Gründen der Entfernung zum Probanden über Skype durchgeführt wurde. Vor Beginn der Evaluation hatten die Probanden keinerlei Detailwissen zu den Inhalten der Befragungen, um eine Voreingenommenheit auszuschließen. Sie wurden lediglich in einer Email über das befragte Themenfeld und die Emendo Werkzeugkette informiert<sup>3</sup> und um die Zustimmung einer Teilnahme an der Studie gebeten.

Die Zeit für eine vollständige Durchführung einer Evaluation betrug im Durchschnitt 1,5 Stunden ( $SD = 0,36h$ ;  $V(x) = 0,13h$ ). Die einzelnen Evaluationen wurden nach Zustimmung durch die Probanden zu Dokumentationszwecken aufgezeichnet. Die nachfolgende Durchführungsbeschreibung wurde in einem Pretest mit zwei Personen vorab validiert.

Auf Basis der methodischen Vorüberlegungen wurde ein zweistufiges Evaluationskonzept entwickelt und wie im Folgenden beschrieben, durchgeführt. Beide Stufen des Konzepts untergliedern sich jeweils in eine inhaltliche Erhebungsphase und Rekonstruktionsphase, gemäß der Erhebung subjektiver Theorien. Die erste Stufe zielte auf allgemeinere Aspekte von Emendo ab, während die zweite Stufe konkret eine angeleitete Anwendung des Designers mit der DSML vorsah.

<sup>3</sup> Es wurde lediglich das Themenfeld genannt, in welchem die Evaluation durchgeführt wurde, den Zweck der Evaluation sowie Emendo als Konzept mit seinen Werkzeugen sehr kurz und oberflächlich in einem Satz vorgestellt.

Pro-band	Alter*	Lehrer-führung	Akad.-Grad**	Fach-bereich***	Dienstl.-Stellung	Einsatz Ga-mification	Einsatzbsp. Gamification	Einsatz LMS	Einsatzbsp. LMS
A	46-55	16-20	Dr.	I	Prof.	eher selten	Greenfoot, Spiele	häufig	Moodle
B	36-45	11-15	Dr.	I	Prof.	selten	TicTacToe etc.	sehr selten	-
C	26-35	< 2	MMD	W	wi. Ma.	eher selten	Quiz, Selbsttests, Rollenspiele	sehr häufig	Moodle
D	26-35	< 2	MMD	P	wi. Ma.	eher selten	Hörsaalspiele, Belohnungen, Wettbewerb	sehr häufig	Moodle
E	26-35	2-5	MMD	I	wi. Ma.	sehr häufig	Belohnungen für Aufgaben	sehr häufig	Moodle
F	56-65	16-20	Dr.	I	Prof.	häufig	Krimi-Rätsel, Quiz, Rateshows, Kryptokämpfe	sehr selten	-
G	36-45	6-10	Dr.	W	wi. Ma.	sehr häufig	Diverse GDEs pro Lehrveranstaltung	sehr häufig	Olat u.a.
H	26-35	2-5	BA	P	wi. Ma.	häufig	Wettbewerb, Auszeichnungen, Preise, Punkte	sehr häufig	Moodle

Tabelle 19: Übersicht über das Sample inkl. Merkmalsausprägungen

\* - in Jahren

\*\* - Bachelor (BA); Master / Magister / Diplom (MMD); Professor (Prof.)

\*\*\* - Informatik (I); Wirtschaftswissenschaften (W); Pädagogik (P)

Die erste Stufe zielte darauf die beiden Forschungsfragen FFE\_1 und FFE\_2 zu beantworten. Aus Gründen der Vergleichbarkeit und Gleichbehandlung wurde jeder teilnehmenden Person zu Beginn ein Video<sup>4</sup> gezeigt, welches die Idee von Emendo, dessen Bestandteile und Funktionsumfang aufzeigte sowie für das Verständnis notwendige Hintergrundwissen zu Themen der Spielifizierung und DSM vermittelte. Dieses Video konnte von Probanden bei Bedarf mehrfach abgespielt werden. Rückfragen an den Forscher waren möglich, obwohl sich dieser zwar im Raum aufhielt, aber während dieses Inputs passiv in einer Beobachterrolle auftrat. Im Anschluss erfolgte ein halbstandardisiertes und leitfadengestütztes Interview zur Informationserhebung. Der dafür erarbeitete Leitfaden enthielt hypothesenungerichtete (HU), hypothesengerichtete (HG) und Stör-Fragen (SF) sowie darüber hinaus Stimuli-Fragen, um das Interview einzuleiten (Scheele u. Groeben, 1988). Tabelle 20 gibt einen Überblick über eingesetzte Fragetypen bzw. Stimuli zu an den Forschungsfragen ausgerichteten Konzepten.

FFE Konzept	Frage
1 -	Wie gefällt Ihnen Emendo? Ist das Konzept für Sie nachvollziehbar? (Stimuli)
1 Einsetzbarkeit	Würden Sie zustimmen, dass Emendo für einen Einsatz in Lehrveranstaltungen geeignet wäre? (HG)
1 Einsetzbarkeit	Was erwarten Sie sich von einem Einsatz in Lehrveranstaltungen? (HU)
1 Einsetzbarkeit	Welche Vor- und Nachteile sehen Sie mit einem Einsatz verbunden? (HU)
2 Rahmenbedingungen	Angenommen Sie stimmen einem Einsatz in Lehrveranstaltungen zu; unter welchen Bedingungen müsste dieser stattfinden? (HG)
2 Rahmenbedingungen	Wäre ihr momentan genutztes LMS nicht auch eine Alternative zur Erstellung und Implementierung von spielifizierten Lehr-Lernarrangements? (SF)

Tabelle 20: Leitfragen für die inhaltliche Erhebungsphase der ersten Evaluationsstufe

<sup>4</sup> Das dreieinhalb-minütige Video ist zu finden unter <https://www.dropbox.com/s/oqelvzkbg1iyaf/EmendoVideo.mp4?dl=0>, abgerufen am 28.05.2018.

Während der ersten Phase wurden die Aussagen von Probanden auf Kärtchen notiert und im Dialog mit den Probanden validiert. Im Anschluss erfolgte die Durchführung der zweiten Phase, in der notierte Kärtchen durch Probanden in eine Struktur gebracht wurden. Außerdem wurde Probanden die Möglichkeit gegeben sogenannte Operator-Kärtchen<sup>5</sup> einzufügen, die entweder aus einem vordefinierten Set oder und den Vorschlägen der Probanden entstammten. Wurde durch den Probanden signalisiert, dass keine weiteren Änderungen oder Ergänzungen an den Kärtchen und deren Struktur vorgenommen werden sollte, wurde eine Fotodokumentation durch den Forscher angefertigt, welche zusammen mit der Audioaufnahme für die Auswertung herangezogen wurde. Abbildung<sup>6</sup> 84 zeigt beispielhaft das dokumentierte Ergebnis der SLT zu dem ersten Evaluationsteil in einer Evaluation mit einem Probanden.

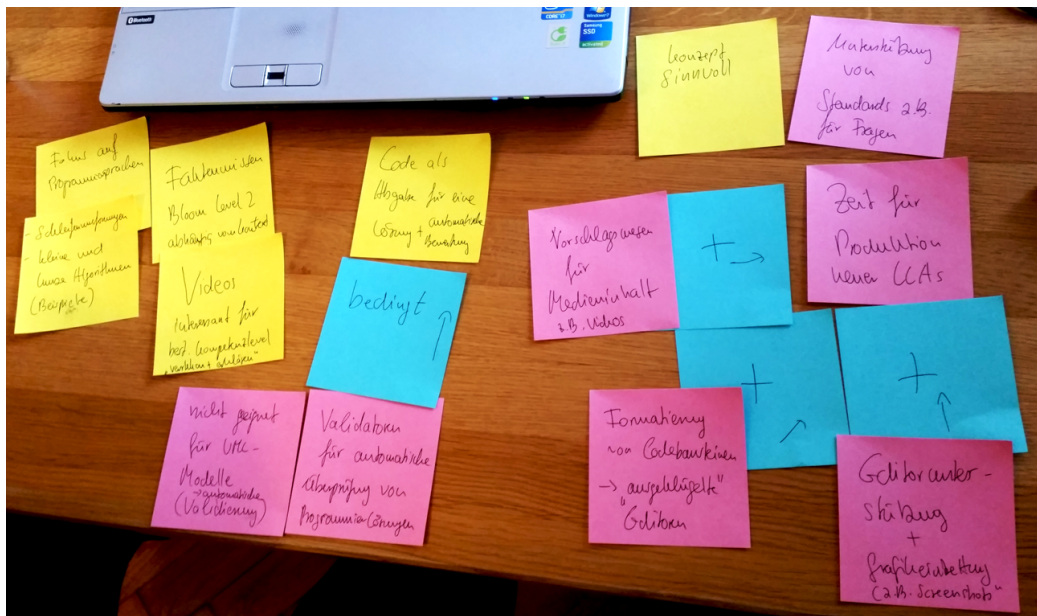


Abbildung 84: Beispielhafte Fotodokumentation eines Ergebnisses zum ersten Evaluationsteil

Bei der zweiten Stufe des Konzepts stand die angeleitete Anwendung des Emendo Designers mit der Emendo DSML im Vordergrund. Diese Stufe zielte darauf ab die restlichen drei Forschungsfragen FFE\_3, FFE\_4, FFE\_5 zu beantworten. Hierfür wurde für die erste Phase den Probanden ein Geheft<sup>7</sup>

5 Dabei handelt es sich um einen Kartentyp, der zwei weitere Karten miteinander in Beziehung setzt. Beispiel: positiv für, bedingt durch, Bedingung für etc.

6 Gelbe Kärtchen stellen allgemeine Aussagen dar, rote Kärtchen Verbesserungsvorschläge und blaue Kärtchen sind Operator-Kärtchen.

7 Siehe Anhang A.8.

ausgehändigt, bestehend aus einem einführenden Text, allgemeinen Hinweisen zur Benutzung des Emendo Designers, einer Szenariobeschreibung, einer Beschreibung der Notation der Emendo DSML sowie einer Verbindungsübersicht für Notationselemente. Die Probanden wurden gebeten das Geheft zu studieren und anschließend das gegebene Szenario umzusetzen. Während der Umsetzung waren Rückfragen an den Forscher möglich. Letzterer war während der Umsetzung passiv, um für teilnehmende Personen nicht das Gefühl einer Testsituation zu verstärken. Während der Durchführung wurde ein Beobachtungsprotokoll erstellt, in dem sämtliche Interaktionen mit der Testperson (inkl. Grund der Interaktion) durch den Forscher festgehalten wurde, um im anschließenden Interview darauf Bezug nehmen zu können. Nach der Umsetzung des Szenarios durch die Probanden folgte wie in der ersten Stufe ein halbstandardisiertes und leitfadengestütztes Interview, dessen Fragen in Tabelle 21 dargestellt sind.

	<b>FFE Konzept</b>	<b>Frage</b>
3	-	Welche Erfahrungen machten Sie beim Modellieren? (Stimuli)
3	Sprache	Wie bewerten Sie die DSML im Allgemeinen? (HU)
3	Sprache	Würden Sie zustimmen, dass die DSML in Bezug auf den Anwendungskontext und auf ihre Elemente nicht vollständig ist? (HG)
3	Sprache	Gibt es Sprachelemente die Sie vermisst haben? Wenn ja, welche und warum? (HU)
4	Sprache	Wie beurteilen Sie die Sprachlogik der DSML? (HU)
4	Sprache	Würden Sie zustimmen, dass die Verständlichkeit der Sprache daran hindert diese schnell zu erlernen? (SF)
5	Designer	Welche positiven und negativen Aspekte sind Ihnen beim Emendo Designer aufgefallen und warum? (HU)
5	Designer	War der Umgang mit dem Emendo Designer für Sie logisch und intuitiv? Bitte begründen Sie. (HU)

Tabelle 21: Leitfragen für die inhaltliche Erhebungsphase der zweiten Evaluationsstufe

In der zweiten Phase wurde, wie bereits in der ersten Stufe der Evaluation beschrieben, die SLT angewandt, um die durch den Forscher dokumentierten Aussagen zu strukturieren und miteinander in Bezug zu setzen. Anschließend hatten die Probanden die Möglichkeit selbst Dinge im Emendo Designer auszuprobieren und das Werkzeug weiter zu explorieren. Dieser Schritt diente der Erfahrungsverdichtung. Daraufhin wurden die Probanden gebeten, sofern die optionale Anwendung in Anspruch genommen wurde, die gesammelten Erfahrungen zu äußern, bevor das Abschlussgespräch begann. In Letzterem wurden Probanden gebeten Anmerkungen einzubringen, die bisher noch nicht zur Sprache kamen und Ihre Einschätzung aus der ersten Phase in Bezug auf die Einsatzmöglichkeiten von Emendo in Lehrveranstaltungen zu überdenken und ihre Einschätzung nach Absolvierung der zweiten Stufe mitzuteilen.

Nach dem Abschlussgespräch wurden Probanden gebeten einen schriftlichen Fragebogen<sup>8</sup> auszufüllen, welcher das Expertenprofil jedes Probanden abfragte. Die dabei erhobenen Daten wurden bereits in der Beschreibung des Samples vorgestellt.

#### 8.2.4 *Ergebnisse*

Für die Auswertung der Daten wurde sowohl auf die Audiomitschnitte und die Fotodokumentationen der einzelnen Gesprächsergebnisse zurückgegriffen. Letztere wurden transkribiert und gemäß den Richtlinien einer strukturierenden Inhaltsanalyse (Mayring, 2010) anschließend kodiert. Die Kodierungen wurden zu Schlüsselkategorien zusammengefasst und in einem nächsten Schritt Letztere den jeweiligen Forschungsfragen (unter Berücksichtigung der Audioaufzeichnungen) zugeordnet. Darüber hinaus wurden für die Kodierungen überindividuell Frequenzanalysen durchgeführt, um die relative Häufigkeiten pro Forschungsfrage zu erfassen. Daraus konnten Indizien für die Wichtigkeit der genannten Aussagen abgeleitet werden (Christmann et al., 1999, S. 148). Zu beobachten war, dass Operator-Kärtchen während den Evaluationen vergleichsweise selten von Probanden eingesetzt wurden. Vielmehr erfolgte eine Strukturierung der Aussagen auf inhaltlicher Ebene, sodass diese Art der Strukturierung auch für die Präsentation der Ergebnisse verwendet werden soll.

---

8 Siehe Anhang A.9.



### 8.2.4.1 Ergebnisse zur ersten Forschungsfrage

Es erfolgte, ausgehend von der Forschungsfrage und den darin verorteten Konzepten, jeweils eine grafische und wertneutrale Aufbereitung der Aussagen von Probanden, orientiert an den Kodierungen, wie in Abbildung<sup>9</sup> 85 zu erkennen. Dies dient der inhaltlichen Übersicht über die jeweiligen Ergebnisse. Im Folgenden werden die Codes pro Schlüsselkategorie näher detailliert. Dabei wurden auf Kärtchen verschriftliche Aussagen nicht interpretiert, sondern werden so wiedergegeben wie diese entweder auf einem Kärtchen zu dokumentiert sind oder wie aus den Audioaufzeichnungen hervorgeht.

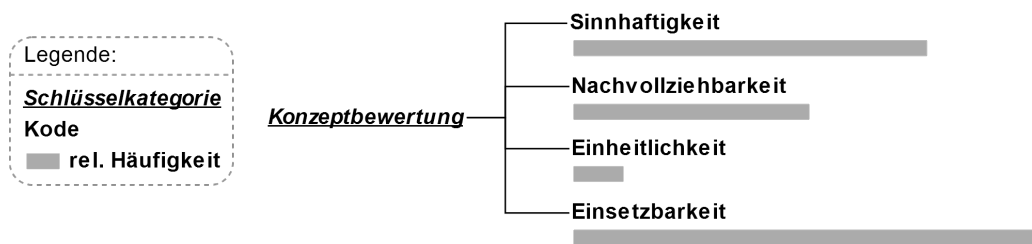


Abbildung 85: Übersicht über die Codes der Stimuli-Frage von FFE\_1

Abbildung 85 stellt die Aussagen in Bezug auf die Stimuli-Frage zu FFE\_1 dar. Ein konkretes Konzept ist hier nicht existent. Probanden äußerten sich vor allem in Bezug auf die Sinnhaftigkeit, Nachvollziehbarkeit, Einheitlichkeit und Einsetzbarkeit von Emendo. Es handelte sich jeweils um bewertende Aussagen, intendiert durch die Fragestellung.

Die Probanden A, E und G bewerten das Gesamtkonzept von Emendo als sinnvoll und zielführend in Bezug auf die Konzeption und Implementierung von spielifizierten Lehr-Lernarrangements. Proband F äußert, dass Emendo als Tool „genial“ sei. Von Proband E wird gesagt, dass ein Einsatz von Emendo in einer Lehrveranstaltung eine Erleichterung wäre. Auf Basis der Bestandteile von Emendo wurde die Unterscheidbarkeit von Lern- und Leistungsaufgaben als positiv bewertet (Proband D) sowie die Möglichkeit des Einsatzes von spielerischen Elementen, da diese Anklang bei Studierenden finden (Proband H). Das Konzept von Emendo wurde als nachvollziehbar beurteilt (Proband D, E, F), wobei eine subjektiv negativ empfundene Konnotation mit dem Begriff der Modellierung geäußert wurde (Proband C). Die Einheitlichkeit der DSML wurde positiv hervorgehoben (Proband F) sowie Emendo eine grundsätzliche Einsetzbarkeit in Lehrveranstaltungen attestiert (Proband D, E). Das Vorhan-

<sup>9</sup> Die relative Größendarstellung der einzelnen Balken unterhalb jedes Codes spiegelt das Ergebnis der überindividuellen Frequenzanalyse im Kontext eines Konzepts zu einer Forschungsfrage wieder. Dies gilt für alle Ergebnisabbildungen im Folgenden, die wertneutral eine Übersicht über die Konzepte, deren Schlüsselkategorien und Codes geben.

densein von Lernpfaden in Emendo wurde zudem als positiv hervor gehoben, da es der Diversität und Heterogenität von Studierenden gerecht werde (Proband H).

Bezogen auf das Konzept der Einsetzbarkeit der ersten Forschungsfrage, kristallisierten sich zwei Schlüsselkategorien heraus, die aus den Codes der einzelnen Aussagen gebildet werden konnten: Einsatzbereiche und Erwartungen an einen Einsatz (siehe Abbildung 86).

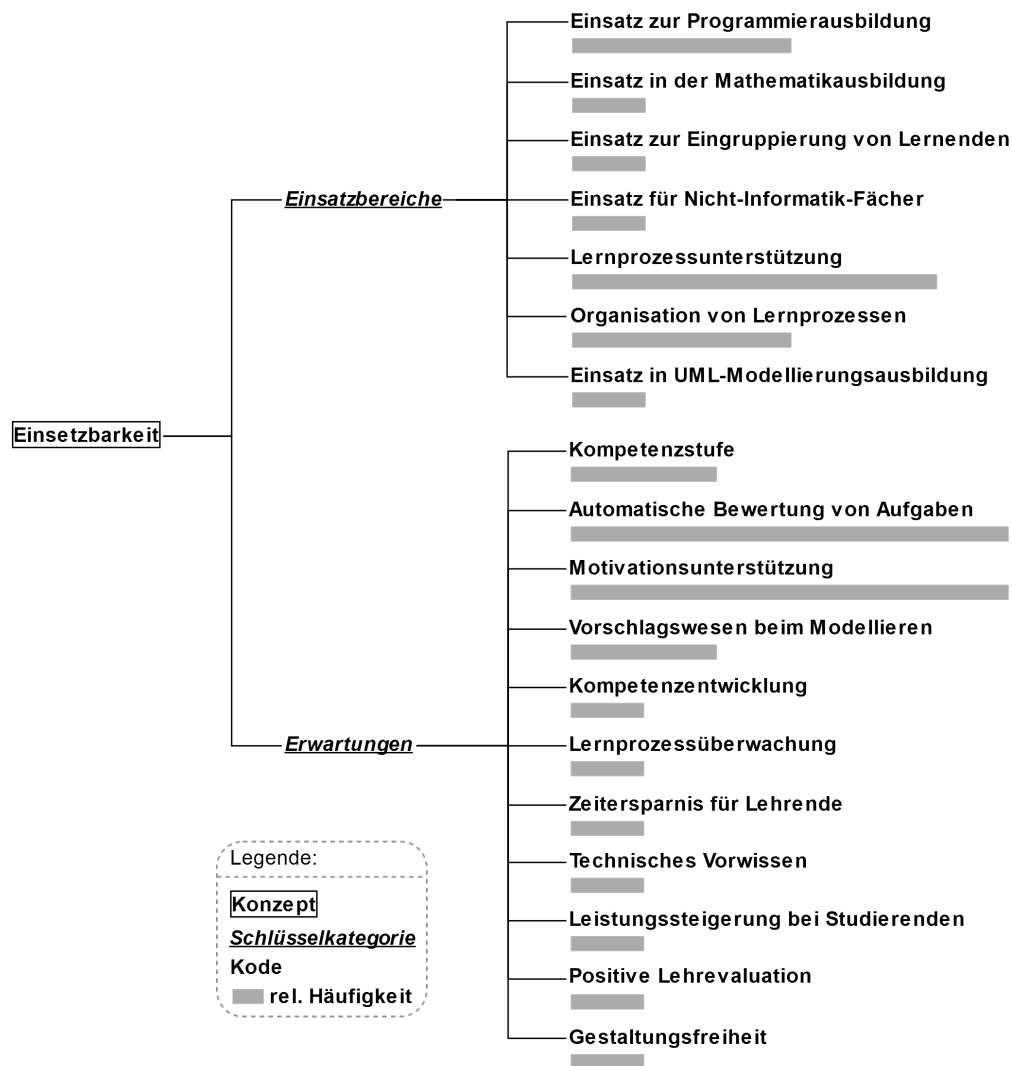


Abbildung 86: Schlüsselkategorien und Codes zum Konzept Einsetzbarkeit der FFE\_1

Proband A sah in Emendo die Einsatzmöglichkeiten in der Programmierausbildung, vor allem bei der Formulierung von kleinen und kurzen Algorithmen in unterschiedlichen Programmiersprachen oder Tätigkeiten, wie das Üben von Schleifenumformungen. Selbiger empfand Emendo als eingeschränkt

einsetzbar für die UML-Modellierungsausbildung, da Emendo keine automatische Validierung von UML-Diagrammen vorsieht. Nach Proband C wäre ein Einsatz von Emendo in der Mathematikausbildung aber auch darüber hinaus möglich, da in Ersterer Lösungen eindeutig überprüfbar seien. Ebenfalls Proband C sieht durch Emendo eine Eingruppierungsmöglichkeit von Studierenden gegeben, in dem damit Eingangstests gebunden an ein Levelsystem durchgeführt werden. Proband F attestiert Emendo eine Einsetzbarkeit, die sich nicht nur auf Informatik-Fächer beschränkt.

Emendo wird als Mittel der Lernprozessunterstützung gesehen, da es den persönlichen Lernprozess visualisiert und damit einen Beitrag zur Individualisierung des Lernens leistet (Proband B). Es sei jedoch fraglich, so Proband C, ob diese Form der Individualisierung sich für jeden Lerntyp eigne. Zudem wurde geäußert, dass die Möglichkeit Feedback zu Lösungen durch Dozierende zu geben und damit Individuallösungen und Beurteilungen zuzulassen, als positiv empfunden wird (Proband C).

Die Möglichkeit für Lernende anreizgestützte Lernpfade zu nutzen (Proband E) und während deren Bearbeitung den Lernfortschritt durch Dozierende zu tracken, wird als positiv beurteilt (Proband F). Mit Emendo bestünde zudem die Möglichkeit eine definierte Lernmenge für Lernende abzubilden, was in Moodle als kompliziert durch Proband C beschrieben wird. Proband G fügt an, dass der Fokus, welcher sich bei Emendo nicht nur auf ein Lernelement erstreckt, sondern auch Gruppen von Lernelementen einschließt, positiv bewertet wird, da so komplexere Lernsettings abgebildet werden können. Proband H gibt an, dass durch die Portabilität des Emendo LMS dies auch innerhalb von Lehrveranstaltungen nutzbar sei, auch für Lehrende, die beispielsweise in situ Lösungen von Studierenden bewerten können.

Die zweite Schlüsselkategorie fokussierte die Erwartungen an einen Einsatz von Emendo in Lehrveranstaltungen. Die häufigste Erwartung der Probanden an Emendo ist die der Motivationsunterstützung für Lernende. Damit assoziiert werden das Bevorzugen von „Karotte vor Peitsche“ (Proband B) und die damit verbundene Erwartung, dass Emendo genügend Anreize zum „aktiven Tun“ schaffen kann (Proband B, G). Darüber hinaus wird durch Emendo eine Lernmotivationssteigerung erwartet, welche sich als Folge auf die Qualität der Aufgabenbearbeitung positiv auswirkt (Proband E). Auch eine Leistungssteigerung und ein höheres Engagement von Studierenden durch den Einsatz von Emendo sei zu erwarten, so Proband G. Dabei könnte Emendo, so Proband E, den Fokus – unter Umständen – der GDEs zu sehr auf extrinsi-

sche Motivation legen. Für Lehrende könnte diese Motivationsunterstützung zu einer besseren Lehrevaluation führen (Proband G).

Fast ebenso wichtig wie die Motivationsunterstützung war für die Probanden aus quantitativer Sicht die automatische Bewertung von Aufgabenlösungen. Letztere Möglichkeit wird von Proband A als vor allem im Kontext der Programmierausbildung als wichtig erachtet. Proband C sieht selbige Wichtigkeit der automatischen Überprüfbarkeit von Lösungen nicht im Programmierkontext, sondern aus Aufgabensicht für den Typ Freitextaufgaben und deren Lösung. Diese Meinung wird von Proband E geteilt, der jedoch die Erwartung einer automatischen Plagiatsüberprüfung von Lösungen an Emendo äußert.

Probanden erwarten eine Unterstützung beim Modellieren mit dem Emendo Designer, beispielsweise in Form eines Vorschlagwesens für Videos zu bestimmten Themeninhalten (Proband A, E). Proband E spricht in diesem Zusammenhang von einer „Multidimensionalisierung von Lerninhalten“. In Bezug auf die Kompetenzentwicklung äußert Proband E die Erwartung, dass Emendo über eine Lehrveranstaltung hinaus eingesetzt werden sollte. Proband A äußert in Bezug auf die Kompetenzentwicklung zudem, dass Emendo mindestens die ersten beiden Kompetenzstufen nach Bloom'scher Kompetenzmatrix (Bloom et al., 1956), also erinnern und verstehen, erfüllen muss. Während dem Lernen bzw. der damit verbundenen Kompetenzentwicklung, soll Emendo Learning Analytics bereitstellen, die ein „rechtzeitiges Eingreifen von Lehrenden“ erlauben (Proband E). Für die Konzeption und Implementierung von spielifizierten Lehr-Lerneinheiten, als Mittel der Kompetenzentwicklung, soll Emendo kein technisches Vorwissen zur Benutzung von Lehrenden erfordern (Proband G), den Lehralltag zeitlich entlasten (Proband E) und dabei Lehrende bei der Konzeption derartiger Lehr-Lernarrangements nicht in ein gestalterisches Korsett wie Moodle zwingen und zu Kompromissen in der Umsetzung zwingen (Proband H).

#### 8.2.4.2 Ergebnisse zur zweiten Forschungsfrage

Die Abbildung 87 zeigt eine Übersicht über die wertneutralen Codes und deren Schlüsselkategorien zu dem Konzept Rahmenbedingungen der zweiten Forschungsfrage FFE\_2.

Die Codes konnten in die Schlüsselkategorien organisatorische, materielle, wissenschaftliche und technische Voraussetzungen geclustert werden. Einen auffallend großen Anteil an Aussagen von Probanden sind dem Code des Zeitfaktors als organisatorische Voraussetzung zuzuordnen. Alle Probanden äußern den zeitlichen Aufwand, um sich in Emendo einzuarbeiten und da-

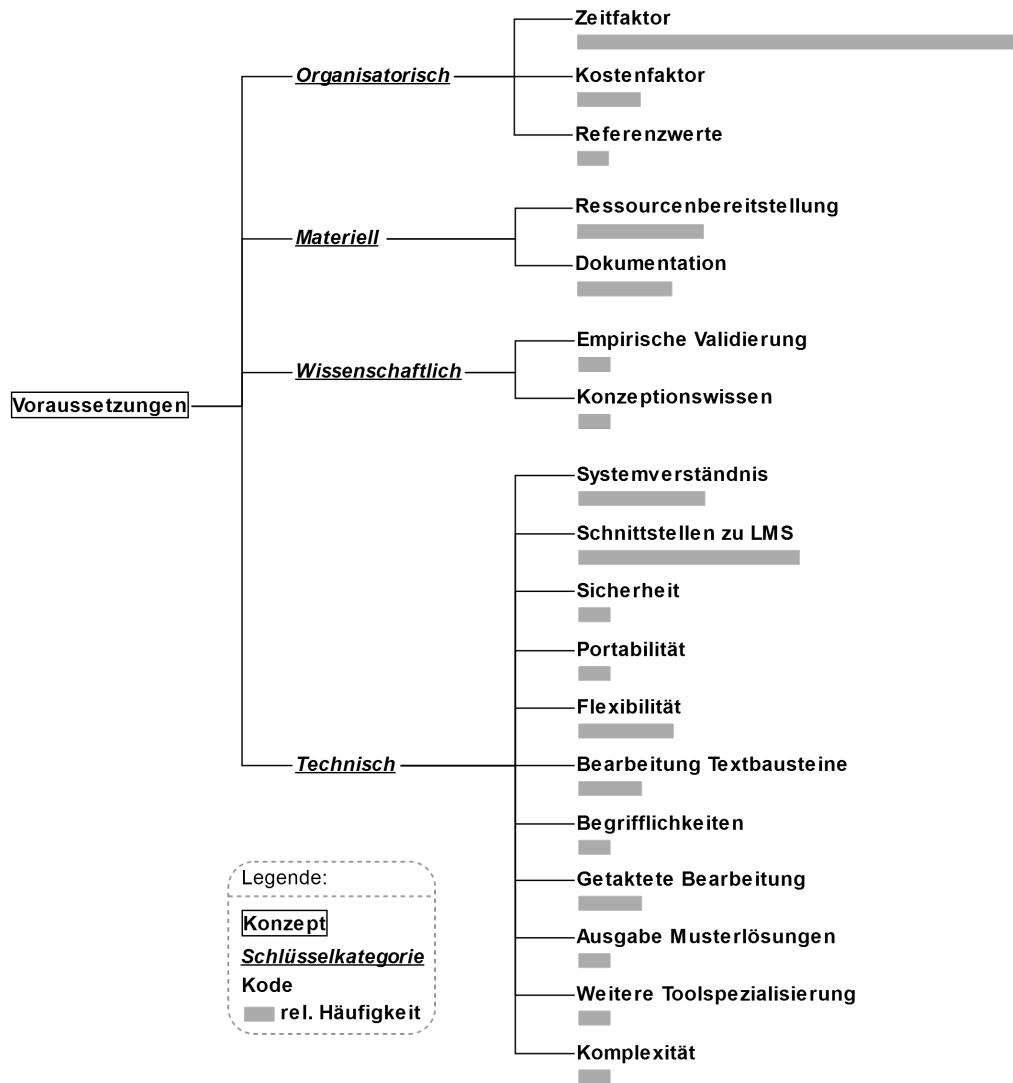


Abbildung 87: Schlüsselkategorien und Codes zum Konzept Voraussetzungen der FFE\_2

mit neue Lehr-Lernarrangements zu erstellen, bestehende einzelne Arrangements oder gesamte Lehrveranstaltungen zu adaptieren, als eine kritische Voraussetzung, angemessen bzw. klein gehalten werden muss (Proband A-H). Auch eine geringe Einarbeitungszeit für Studierende wird von Proband E als Voraussetzung formuliert. In puncto Kosten als eine weitere organisatorische Voraussetzung, sollte Emendo kostenlos (Proband E) bzw. den Kosten angemessen sein (Proband C) und Referenzwerte im Einsatz von Dozierenden und Studierenden bereitstellen, um Emendo mit bestehenden LMS, wie Moodle, vergleichen zu können (Proband C). Als materielle Voraussetzungen wurde eine Bereitstellung von Ressourcen genannt, vor allem in Bezug auf vordefinierte Fragensets im Designer (Proband A), didaktisch fundierte Kombinationen aus Lernelementen (Proband C) sowie eine vorsortierte Mediensamm-

lung „erfolgreicher“ Videos bzw. Podcasts zu einem bestimmten Lerninhalt (Proband D). Zudem besteht Bedarf an einer Dokumentation und Support für die Verwendung des Designers (Proband C), idealerweise mit Anleitungen und Videotutorials (Proband E, G).

Aus wissenschaftlicher Sicht soll Emendo Validierungen durchlaufen haben, die zeigen, dass ein Einsatz tatsächlich zu einer Motivationssteigerung bei Studierenden führt (Proband C), sowie dabei ein geringes Maß an Detailorientierung für die Erstellung von spielifizierten Lehr-Lernarrangements voraussetzen (Proband C).

Aus technischer Perspektive, sehen die Probanden C, D, G und H eine Voraussetzung für den Einsatz von Emendo in der intuitiven und selbsterklärenden Anwendung v.a. des Emendo Designers, um zu einem „schnellen Ergebnis zu kommen“. Dafür soll Emendo Begriffe und Bezeichnungen verwenden, die nicht technisch sind (Proband C), und die Komplexität der Verwendung aufgrund der Teilkomponenten der Werkzeugkette gering halten (Proband H). Ebenso wird die Schaffung von Schnittstellen zu bestehenden LMS, wie Moodle thematisiert (Proband C, D, E, G), auch mit Ziel, Studierenden „kein weiteres Tool zumuten zu müssen“ (Proband H). Emendo soll Plattform-unabhängig eingesetzt werden können und aktuelle Sicherheitsanforderungen erfüllen (Proband B). Es soll zudem flexibel und erweiterbar in Bezug auf die Elemente der DSML sein und damit keine GestaltungsKompromisse von Lehrenden fordern sowie auch ohne Game Design-Elemente einsetzbar sein (Proband H). Während der Konzeption soll vor allem der Designer Lehrende mit ausgeklügelten Editoren zur Textbausteinbearbeitung unterstützen, um beispielsweise Codebausteine automatisch zu formatieren oder Screenshots in Texte einzubetten (Proband A). Darüber hinaus wäre die Ausgabe von Musterlösungen für Studierende über das Emendo LMS eine Voraussetzung (Proband D), ebenso wie die einheitlich getaktete Bearbeitung von Lernarrangements im Emendo LMS, mit idealerweise eingebetteter Präsentation von PowerPoint-Folien (Proband F).

#### 8.2.4.3 *Ergebnisse zur dritten Forschungsfrage*

Bei der Stimuli-Frage zu FFE\_3 äußerten sich Probanden hauptsächlich in Bezug auf den Modellierungserfolg im vorher durchgeführten Szenario und zur automatischen Umsetzung von modellierten Lerneinheiten (siehe Abbildung 88). Probanden D, F und G äußerten, dass sich das vorgegebene Szenario gut mit dem Emendo Designer und der darin eingebetteten DSML umsetzen ließe und dabei alle Szenariobestandteile umgesetzt werden konnte. Proband D hob zudem heraus, dass man „sich durch das Generieren keine Sorgen um die

Darstellung machen“ müsse. Weitere bewertende Aussagen zielten direkt auf die Sprache oder auf den Designer ab, weshalb sie nicht in den Ergebnissen der Stimuli-Frage erfasst wurden.

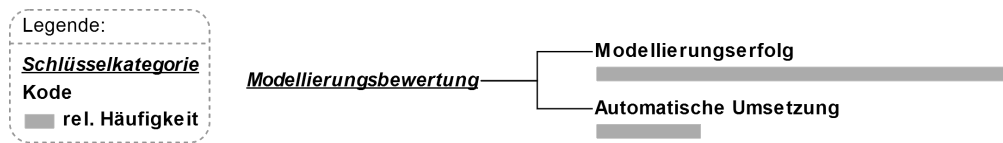


Abbildung 88: Übersicht über die Codes der Stimuli-Frage von FFE\_3

Bezogen auf die Sprache ließen sich die Schlüsselkategorien Sprachumfang und Notation aus den Codes der Aussagen verdichten (siehe Abbildung 89). In Bezug auf die Vollständigkeit der Sprache, wurde der Sprachumfang als vollständig für den vorgesehenen Anwendungskontext beurteilt (Proband A, B, D, E, G, H). Einzelne Aufgabentypen wurden als fehlend geäußert: Sortieraufgaben, Wimmelbildaufgaben, Zuordnungsaufgaben (Proband F); Zuordnungsaufgaben mit Paramtereinstellung für Zeitdruck (Proband E) sowie ein Aufgabentyp, welcher beispielsweise das „Abhaken von einem gehaltenen Referat von Student x“ erlaube und für einen Blended-Learning-Ansatz von Vorteil wäre (Proband H).

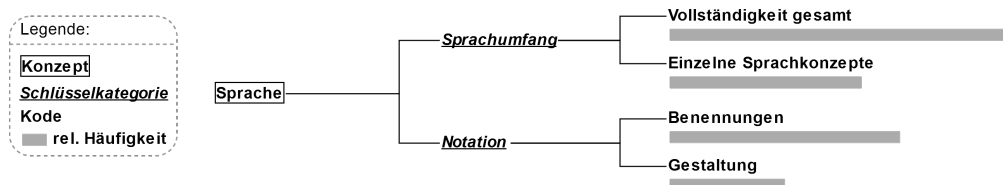


Abbildung 89: Schlüsselkategorien und Codes zum Konzept Sprache der FFE\_3

In Bezug auf einzelne Sprachkonzepte wurde die Verfügbarkeit von Gruppen als Gliederungsmöglichkeiten positiv geäußert (Proband A). Auch die Auswahl an GDEs wurde positiv hervorgehoben, ebenso wie die Verfügbarkeit des Elements Feedback als GDE (Proband G). Von Proband C wurde gesagt, dass das Element Feedback neben einer textuellen Repräsentation multimedialer, „z. B. in Form von kurzen Audiofiles“ verfügbar sein sollte. Darüber hinaus sei ein zusätzlicher Operator für Optionalverbindungen notwendig, der festlegt, welches Lernelement zwingend zu bearbeiten sei (Proband B). Die Benennungen, die in der Notation der Sprache verwendet werden, wurden als zu technisch beurteilt (Proband C, E, G, H). Die Namensgebung der Pflichtverbindung wurde von Proband A als nicht intuitiv bezeichnet. Ebenso wurde kritisiert, dass die deutsche und englische Sprache in der Notation verwendet würden (Proband B). Die Gestaltung der Icons wurde als positiv

und intuitiv beurteilt (Proband A, H), obwohl die Notationselemente von Single Choice und Multiple Choice-Aufgaben nicht gut unterscheidbar wären anhand der Icons (Proband F).

#### 8.2.4.4 Ergebnisse zur vierten Forschungsfrage

Bezogen auf das Konzept Sprache der vierten Forschungsfrage konnten 3 Schlüsselkategorien identifiziert werden (siehe Abbildung 90). Zu der Schlüsselkategorie Verständlichkeit wurden die Aussagen zu zwei Codes verdichtet. Von Proband C, D und G wurde die Anwendung der Sprache als intuitiv bezeichnet. Eine Ausnahme für die intuitive Anwendung stelle die Definition von Bedingungen von Regeln dar (Proband D). Proband G merkt an, dass Modellierungsvorwissen hilfreich für einen intuitiven Umgang mit der Sprache hilfreich sei.

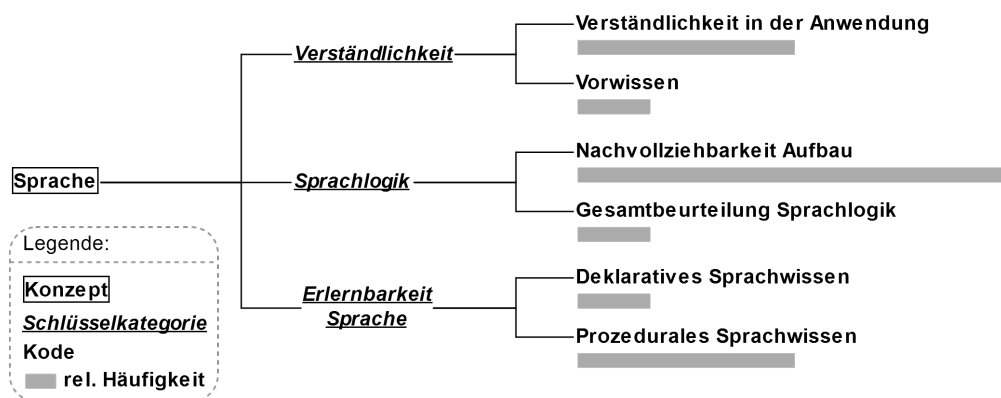


Abbildung 90: Schlüsselkategorien und Codes zum Konzept Sprache der FFE\_4

Bezogen auf die zweite Schlüsselkategorie Sprachlogik äußerten die Probanden A, B, C, D, G und H dass die Sprache einem logischen und nachvollziehbaren Aufbau folge. Auch die Gesamtbeurteilung der Sprachlogik fiel positiv aus (Proband F). Die Aussagen zu der Erlernbarkeit der Sprache wurden zu den Codes deklaratives und prozedurales Sprachwissen verdichtet. Erstere bezeichnet das Wissen um die Sprachelemente und deren Bedeutung, während zweiteres das Anwendungswissen repräsentiert. Das für das Verständnis notwendige deklarative Sprachwissen sei leicht zu erlernen (Proband D), während das in puncto prozedurales Sprachwissen geäußert wurde, dass der Einsatz von Gruppen für fortgeschrittenes Stadium der Benutzer seien sowie die ersten Durchläufe in der Definition von Bedingungen als Teil von Regeln die „größte Hürde“ in der Verwendung darstellen (Proband D). Bezogen auf die allgemeine Anwendung der Sprache äußerte sich Proband H ähnlich. Er sagte,



dass die Erlernbarkeit der Sprache positiv sei unter der Prämisse „man muss es halt Mal gemacht haben, dann weiß man es“.

#### 8.2.4.5 *Ergebnisse zur fünften Forschungsfrage*

In der fünften und letzten Forschungsfrage wurde als zentrales Konzept der Designer fokussiert. Es kristallisierten sich in der Phase der Auswertung 4 Schlüsselkategorien heraus (siehe Abbildung 91). Die erste Schlüsselkategorie fokussiert die GUI des Designers. Eine Gesamtbeurteilung der GUI erfolgte von den Probanden E, G und H. Sie beurteilen die GUI als übersichtlich und nachvollziehbar gestaltet. Dies sei der Fall, da „alle notwendigen Informationen sofort verfügbar [sind]“ und „man muss nichts aufklappen“ (Proband G). Von Proband A und H wurde geäußert, dass Diagramme groß werden könnten und damit die „Ansichtskomplexität“ gesteigert werden würde. Proband H sagte zwar, dass in diesem Fall die Outline hilfreich ist, nannte jedoch als Verbesserungsvorschlag, dass Modelle auf unterschiedlichen Detaillierungsstufen modelliert werden könnten und abhängig von der Stufe nur eine Teilmenge an Elementen angezeigt werden. Die Zeichenfläche, auf der Modellelemente platziert werden, wurde positiv hervorgehoben. Verbesserungsmöglichkeiten sahen Probanden in der Gestaltung der Dialoge zum inhaltlichen Detaillieren der Modellierungselemente, vor allem, dass die Parameterliste der Bedingungsvariablen nicht der 7+-2 Regel (Miller, 1956) folge und zu lang sei (Proband F) sowie Icons für die Detaillierung der Regeln Icons sinnvoll wären, jedoch vom gleichen Probanden angemerkt wurde: „keine Ahnung wie die aussehen“ (Proband G). Darüber hinaus beurteilte Proband C die Flexibilität des Designers in Bezug auf die Anpassungsfähigkeit der GUI positiv.

Die Stabilität bei der Verwendung des Designers (Proband A) sowie dessen Plattformunabhängigkeit (Proband C) wurden positiv herausgestellt und damit Softwareeigenschaften genannt. Für die dritte Schlüsselkategorie Funktionalität waren neben einer nicht vorhandenen Mediensammlung als Unterstützung bei der Modellierung (Proband A) die Benutzereingaben im Designer von Relevanz. Proband A und C äußerten beispielsweise, dass die getrennten Eingaben von Lösungsmöglichkeiten zu Single und Multiple Choice-Aufgaben nachteilig seien bzw. zu viel Aufwand erfordern würden. Weiterhin wurde kritisch geäußert, dass Textfelder in Dialogen keine Formatierungsmöglichkeiten bieten (Proband A), die Einblendung von Modellelementen zu spät erfolge, die auf der Mausposition eingefügt werden können (Proband B), Tooltips zu den Notationselementen und deren Bedeutung hilfreich wären (Proband D, G), sowie Hinweise für die Platzierung von Modellelementen

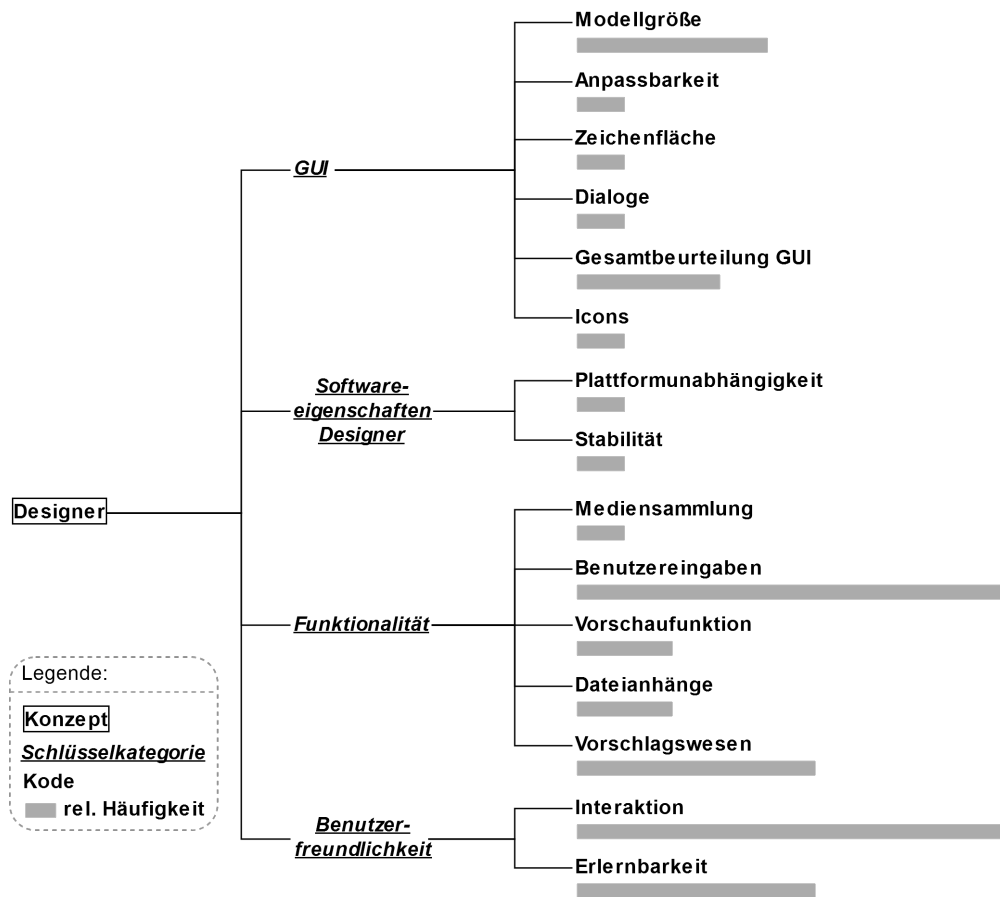


Abbildung 91: Schlüsselkategorien und Codes zum Konzept Designer der FFE\_5

innerhalb oder außerhalb einer Gruppe von Vorteil wären (Proband H). Die Hilfslinien auf der Zeichenfläche für die Positionierung der Elemente wurden als hilfreich beschrieben (Proband B) sowie die Möglichkeit Code in Textfeldern einzubetten wurde positiv hervor gehoben (Proband D). Von Proband C und D wurde vorgeschlagen eine Vorschaufunktion in den Designer zu integrieren, um direktes Feedback darüber zu erhalten, wie ein Modellelement im Emendo LMS dargestellt werde. Außerdem wurde kritisiert, dass der Designer keine Möglichkeit biete, PDF-Dateien anzufügen und für die Benutzung im LMS hochzuladen (Proband C, D). Darüber hinaus äußert Proband B, dass die Vorschläge für die Konnektoren zwischen Elementen sinnvoll seien. Proband E kritisiert jedoch, dass der Designer eine automatische Integration von Lernmaterialien aus dem Internet zu einem Thema bereitstellen solle und bei der Modellierung einer Single Choice-Aufgabe automatisch Antwortmöglichkeiten vorschlagen solle. Ebenso wird von Proband E geäußert, dass Shortcuts für das Einfügen von Modellelementen hilfreich wären und der Designer diese, für den Benutzer änderbar, vorschlagen solle. Ebenfalls soll der Designer „vordefinierte und häufig benutzte Regeln für den Benutzer anbieten“ (Pro-

band G). In Sachen Benutzerfreundlichkeit beschreiben Proband A, E, F und H den Umgang mit dem Designer als intuitiv. Ebenfalls intuitiv sei das Detaillieren der Modellelemente über Dialoge (Proband D, E), auch speziell im Bezug auf die Dialoge der Regeln (Proband E). Proband F merkt an, dass der Designer sehr anschaulich sei und sagt „ich weiß nicht, wie man es besser machen könnte“. Proband G merkt an der „Designer ist funktional gehalten“ und empfindet es jedoch als nicht intuitiv, dass nicht das gesamte Grid für die Platzierung der Modellelemente nutzbar ist, sondern eine Unterteilung in Elemente stattfindet, die innerhalb und außerhalb einer Gruppe liegen können. Von Probanden A und F wird der Umgang mit dem Designer als schnell erlernbar beschrieben. Proband C äußert als Vorbedingung für den Umgang mit dem Designer das Vorhandensein einer technischer Affinität. Proband D verstärkt die Äußerung von Proband C in Bezug auf die Vorbedingung und schließt die Verwendung für nicht-Informatiker aus wegen der Komplexität der Bedingungsdefinition von Regeln. Abhilfe für diese Komplexität könnten Tutorials oder Patterns für die Benutzung des Designers schaffen (Proband E).

#### 8.2.5 Diskussion

Die Ergebnisse zeigen ein überwiegend positives Bild der Probanden von Emendo. Es wird deutlich, dass Emendo Anklang für den Einsatz in der Lehre findet, aber es zeigen sich auch Verbesserungspotentiale, die entweder für einen Prototyp in der Konzeption nicht vorgesehen waren oder bestehende Eigenschaften bzw. Funktionalitäten betreffen. Im Folgenden sollen für jede Forschungsfrage die zentralen Ergebnisse aufgegriffen und diskutiert werden.

##### 8.2.5.1 FFE\_1: *Wie schätzen Lehrende die Einsetzbarkeit von Emendo in Lehrveranstaltungen ein?*

Grundsätzlich zeichnet sich hier ein deutlich positives Bild der Probanden auf eine Einsetzbarkeit in der Lehre. Probanden bewerten das Konzept von Emendo positiv, erwarten durch den Einsatz von Gamification, dass es Anklang bei Lernenden findet und geben an, dass es durch das Vorhandensein des Konzepts Lernpfade der Diversität von Studierenden in Lehrveranstaltungen gerecht werde. Ein gewisses Maß an Skepsis herrscht gegenüber der Modellierung an sich, da diese mit technischen (Vor-) Wissen konnotiert wird. Der DSM-Ansatz hat jedoch zum Ziel, die technische Sicht auszublenden und Benutzern ein Modellierungswerkzeug zur Verfügung zu stellen, welches die

Konzepte der Domäne und deren Anwendbarkeit hervorhebt, die Benutzern bereits bekannt sind. In Bezug auf mögliche Einsatzbereiche von Emendo in der Lehre, zeigen die Ergebnisse eine leichte Tendenz in Richtung von Fachdisziplinen, in denen Aufgaben gestellt werden, die eine deterministisch überprüfbare Lösung besitzen. Diese Überprüfung solle automatisch erfolgen, um Lehrende zu entlasten. Nichtsdestotrotz wird Emendo auch eine Einsatzmöglichkeit in nicht-Informatik-Fächern attestiert, was das zuvor genannte Argument des Einsatzkontextes aufweicht. Obwohl Emendo nicht per se auf einen Fachbereich beschränkt ist, so ist eine gewisse Orientierung auf technisch-orientierte Fachbereiche vorhanden. Dies zeigt sich unter anderem daran, dass Möglichkeiten des Syntax-Highlightings für Quellcode integriert wurden, die in der Regel in einem Fachbereich, wie der Wirtschaftswissenschaft, bis auf ein paar wenige Ausnahmen (z. B. Programmierung von Excel-Makros) keine große Rolle spielen dürften. Die Einbettung von Quellcode ist jedoch optional und muss nicht genutzt werden.

Unabhängig vom Fachbereich wird Emendo von Probanden als ein Werkzeug zur Lernprozessunterstützung wahrgenommen, was zu einer Kompetenzentwicklung von Studierenden beiträgt und deren Motivation durch spielifizierte Elemente anzuregen vermag. Aus Lehrendensicht und bezogen auf die Modellierung von spielifizierten Lehr-Lernarrangements sowie deren Implementierung in Lehr-Lernsettings, scheint der Zeit- bzw. Aufwandsaspekt für die Benutzung von Emendo eine wichtige Rolle zu spielen. Dieser solle, vor allem im Vergleich mit bisherigen LMS, möglichst gering ausfallen und Lehrende beispielsweise bei der Modellierung durch ein ausgeklügeltes Vorschlagswesen in Bezug auf Lernmaterialien oder häufig benutzte Konstellationen von Modellelementen unterstützen. Aktuell leistet Emendo dieses vor allem inhaltlich orientierte Vorschlagswesen bis auf kontextabhängige Selektierungsmöglichkeiten in Dialogen nicht. Dies stellt also eine Verbesserungsmöglichkeit dar, die für eine etwaige Weiterentwicklung von einem Prototypen hin zu einem Produkt von Relevanz wäre. Eine weitere Weiterentwicklungsmöglichkeit ist die Implementierung von Validatoren, die beispielsweise als separate Sprachelemente der Emendo DSML hinzugefügt werden könnten. Damit wäre abhängig vom Aufgabentyp eine Überprüfung von Lösungen von Lernenden automatisiert. So könnte beispielsweise ein Validator implementiert werden, der eine Programmierlösung gegen zuvor definierten Unit-Tests testet oder UML-Modelle statischen Analysen unterzieht. Aktuell ist die Funktionalität zur Validierung einer Lösung mit der Aufgabe an sich verknüpft. Derartige Erweiterungen wären vergleichsweise einfach zu implementieren, da alle Bestandteile der Werkzeugkette entsprechende Abstraktionen dafür bereitstel-

len, um die Gestaltungsfreiheit bei der Modellierung und Implementierung von spielifizierten Lehr-Lernarrangements nicht einzuschränken – eine Eigenschaft, die als Erwartung an Emendo geäußert wurde.

#### 8.2.5.2 FFE\_2: Unter welchen Rahmenbedingungen erfolgt dieser Einsatz?

Die Ergebnisse zu dieser Forschungsfrage zeigen Voraussetzungen die organisatorischer, materieller, wissenschaftlicher oder technischer Art sind. Dominiert aufgrund der Anzahl der Nennungen ist der bereits in der Diskussion zur ersten Forschungsfrage erwähnte Zeitfaktor. Darüber hinaus äußerten sich alle Probanden zu dieser Voraussetzung und nannten diesen als einen kritischen Faktor für die Einsetzbarkeit von Emendo. Außer Frage steht, dass es einer gewissen Einarbeitungszeit seitens Lehrenden und Lernenden in die von ihnen zu verwendenden Werkzeuge bedarf. Bezogen auf die Benutzergruppe der Lehrenden sollte die Konzeption der DSML eine schnelle Erlernbarkeit ermöglichen. Auch der Designer wurde im Rahmen der Entwicklung visuell entsprechend entschlackt und nicht verwendete Funktionen aus der GUI entfernt. Dies sollte ebenfalls dazu beitragen, dass Lehrende sich in die Verwendung der Werkzeuge entsprechend schnell einfinden. Zudem wurde als Unterstützung vor der Modellierung mit dem Designer und der darin eingebetteten DSML der GCDP und der GCDR entwickelt und validiert, was eine systematische Vorstrukturierung der zu modellierenden spielifizierten Lehr-Lerneinheit erlaubt. Allerdings kann der Aufwand zur Adaption bereits bestehender Lernmaterialien in Emendo dadurch nur bedingt verringert werden. Denkbar wäre, Emendo durch einen Importer zu ergänzen, der beispielsweise in einem bestehenden LMS eingepflegte Inhalte nach Emendo überführt. Derartige Funktionalität würde trotz einer Teilautomatisierung händische Nacharbeit erfordern, die aufgrund der Heterogenität von Dateiformaten in denen Lernmaterialien vorliegen oder in einem bestehenden LMS abgelegt sind, zustande kommt. Mit dieser zusätzlichen Funktionalität einhergehend, könnten Lernmaterialien nicht nur für einzelne Instanzen von Emendo gesammelt werden, sondern zentral über einen Server verschiedenen Emendo Instanzen fachübergreifend bereitgestellt werden. Beispielsweise könnte es sich dabei um bereits erstellte Modelle handeln oder Lernmaterialien, die in Modellen eingebettet sind. Als eine weitere materielle Voraussetzung für einen Einsatz von Emendo, wurde die Verfügbarkeit von einer entsprechenden Dokumentation für die Benutzung des Systems genannt. Diese Dokumentation ist aktuell nicht existent, was durch den Prototypcharakter der Anwendung begründet ist. Für den Schritt hin zu einem Produkt wäre diese essentiell, um eine effektive Benutzung weiter zu unterstützen. Was die wissenschaftlichen Vorausset-

zungen angeht, so existieren zahlreiche Validierung, die zeigen, dass Gamification einen positiven Einfluss auf die Motivation haben kann. Ob sich diese Erkenntnis im Falle von Emendo ebenfalls bestätigen lässt, dazu bedarf es weiterer empirischer Untersuchungen, vor allem mit der Benutzergruppe der Lernenden. Das notwendige Konzeptionswissen für die Modellierung von spielifizierten Lehr-Lernarrangements kann zum einen durch eine entsprechende Dokumentation geleistet werden, als auch durch die bereits für die Modellierung erstellten Hilfestellungen durch den GCDP und GCDR. Aus technischer Sicht wurden Schnittstellen zu bestehenden LMS, wie Moodle, als wichtiges Kriterium angeführt. Neben dem bereits erwähnten Importieren von Lerninhalten aus einem LMS nach Emendo, um die Adaption der Lerninhalte für Emendo zu unterstützen, können Ergebnisse oder Bearbeitungsstände, die vor allem im Emendo LMS dokumentiert sind, in ein bestehendes LMS exportiert werden. Detaillierte Möglichkeiten zeigt das Subkapitel 9.2.3 auf. Als weitere technische Voraussetzungen wurde angeführt, dass Emendo technisch Benutzer derartig unterstützen soll, dass das Verständnis des Systems sowie dessen Verwendung schnell gesteigert wird. Auf Lehrendenseite wurden diverse Maßnahmen unternommen, dies zu erreichen. Zum einen in der Konzeption der DSML und darin bekannter Konzepte aus der Domäne, die Entwicklung der konkreten Notation unter Einbezug potentieller Nutzer sowie ein möglichst einfach zu verwendender und übersichtlich gestalteter Modellierungseeditor. Das Kriterium der Flexibilität in der Modellierung von spielifizierten Lehr-Lernarrangements ist ebenfalls erfüllt, da dies durch die bereits mehrfach erwähnten Erweiterungsmöglichkeiten aller Bestandteile der Werkzeugkette gegeben ist. Somit können beispielsweise ausgeklügeltere Editoren in den Dialogen des Designers integriert, verwendete Begrifflichkeiten verändert oder neue Aufgabentypen in die DSML hinzu definiert und entsprechend in die Werkzeuge implementiert werden. Da Emendo auf Java-Technologie basiert, ist eine Portabilität in andere Ausführungsumgebungen gegeben. Sicherheitsaspekte wurden im Prototypen nur rudimentär berücksichtigt. Im Falle einer Weiterentwicklung von Emendo zu einem Produkt muss sich beispielsweise Fragen der technischen Absicherung in puncto Datenschutz und Datensicherheit gewidmet werden, da personenbezogene Daten, vor allem im Emendo LMS erhoben und verarbeitet werden.

#### 8.2.5.3 FFE\_3: *Wie wird der Sprachumfang der Emendo DSML von Lehrenden beurteilt?*

In Bezug auf den Sprachumfang der Emendo DSML äußerten sich fast alle Probanden positiv und beschreiben diesen als für den Kontext vollständig.

Auch wenn einzelne Aufgaben- (z. B. Sortieraufgaben oder Wimmelaufgaben), Feedback- oder Operatortypen genannt wurden, die aktuell in Emendo nicht abgebildet sind, so kann eine nachträgliche Definition in der Sprache durch Erweiterung des Metamodells und der Notation sowie anschließender Implementierung in die Werkzeuge erfolgen. Auch die teils kritisierten zu technischen Benennungen, die in der Notation der DSML und den Dialogen des Designers verwendet werden, können problemlos angepasst werden. Diese Kritik ist allerdings unter dem Gesichtspunkt zu bewerten, dass diese deutlich überwiegend von Probanden stammt, die in der Wirtschaftswissenschaft und Pädagogik beheimatet sind (3 von 4 Probanden).

#### 8.2.5.4 *FFE\_4: Wie werden die Selbsterklärungsfähigkeit der Bestandteile und deren Verwendung der Emendo DSML von Lehrenden eingeschätzt?*

Die Selbsterklärungsfähigkeit und Verständlichkeit der Modellierungssprache werden überwiegend positiv beurteilt. Die geäußerte Kritik bezogen auf die Definition von Bedingungen innerhalb einer Regel stammt von einem Proband des Fachbereichs Pädagogik und ist unter diesem Gesichtspunkt zu betrachten. Die Sprachlogik und die Nachvollziehbarkeit der Sprache wurde von fast allen Probanden positiv beurteilt. Eine schnelle Erlernbarkeit mit Ausnahme der Regeldefinition wurde der DSML durch die Probanden attestiert. Die Sprache spiegelt aus der Domäne bekannte Konzepte wieder, wodurch diese fast durchweg positive Einschätzung der von Selbsterklärungsfähigkeit und Verwendung erklärt werden kann. Nachbesserungsbedarf fordern die Konzepte Gruppe und Regel. Bei dem Konzept Gruppe könnte die von einem Probanden fehlende Verständlichkeit in ersten Modellierungsversuchen im Designer durch die Einführung von Tooltips entgegenwirken. Letztere könnten Hinweise darüber geben, wie Konzepte zu verwenden sind und welche semantische Bedeutung sie besitzen. Bezogen auf die Definition von Bedingungen von Regeln, könnte eine andere Darstellungsform der Dialoge Abhilfe für die Hürde in der Verwendung schaffen. Möglicherweise ist die rekursive Struktur der Dialoge nicht intuitiv genug. Jedoch ist auch hier anzumerken, dass die Kritik von einem Proband stammt, der dem Fachbereich Pädagogik zuzuordnen ist. Dem gegenüber gestellt steht eine Aussage eines Probanden aus der Informatik, welcher die Definition von Regeln über die Dialoge als erwähnenswert positiv beschreibt.

#### 8.2.5.5 FFE\_5: *Wie schätzen Lehrende die Funktionalität des Emendo Designers zum Modellieren mit der Emendo DSML ein?*

Bezogen auf die GUI herrscht ein überwiegend positives Meinungsbild in Bezug auf den Designer vor. Probanden beschreiben das Werkzeug als übersichtlich und nachvollziehbar gestaltet. Kritisiert wurde die Größe der Diagramme, was die Ansichtskomplexität für Benutzer erhöhen könnte. Als Verbesserungsmöglichkeit wurde hierzu eine Strukturierung der Modelle in unterschiedliche Ebenen geäußert. Dies hätte zur Folge, dass beispielsweise die Modellierung in mehrere Schritte unterteilt werden könnte, die jeweils einer Detaillierungsebene entsprechen. Denkbar wäre in der oberste Detaillierungsebene alle Lernbausteine zu definieren, die für ein Lehr-Lernarrangement eingesetzt werden sollen. In einer zweiten Ebene, könnte die Gruppierung der Lernbausteine stattfinden und eine Anreicherung mit Regeln. Und in einer dritten und letzten Ebene das Einfügen von Verbindungen, die Lernbausteine in Lernpfade anordnen. Aus technischer Sicht wäre vor allem eine Anpassung des Designers zu leisten. Das für die Erstellung des Designers verwendete Sirius Framework unterstützt die Möglichkeit unterschiedliche Sichten (Viewpoints) auf ein Metamodell einzunehmen. Die beschriebenen Ebenen könnten auf Basis dieser Viewpoints in Sirius umgesetzt werden. Weiterhin könnten einzelne Dropdown-Listen in Dialogen des Designers unter Berücksichtigung der 7+-2-Regel angepasst werden. Auch könnte eine Überarbeitung der Dialoge dahingehend erfolgen, dass Icons eingefügt werden, welche die Selbstbeschreibungsfähigkeit und damit die Gebrauchstauglichkeit steigern. Die Softwareeigenschaften des Designers wurden positiv beurteilt. Die Plattformabhängigkeit ist der Java-Technologie geschuldet während die Stabilität ein Ergebnis der Maßnahmen zur Steigerung der Fehlertoleranz im Designer darstellt. Aussagen zu Benutzereingaben im Designer stellen den größten Anteil an der Schlüsselkategorie Funktionalität dar. Marginale Änderungen, wie die Definition von Antwortmöglichkeiten innerhalb der Dialoge zu Single und Multiple Choice-Aufgaben sind im Designer vergleichsweise einfach umzusetzen. Auch ein Einfügen von Tooltips zu Notationselementen, die beispielsweise Hinweise zur der Platzierung von Modellelementen auf der Zeichenfläche liefern, erscheint sinnvoll. Aktuell noch fehlende Formatierungsmöglichkeiten von Texten in den Dialogen des Designers sind noch nachzurüsten. Hierfür können beispielsweise Hypertext Markup Language (HTML)-basierte Editoren für die Textfelder verwendet werden. Dies hätte allerdings eine Änderung des Generators zur Folge, da einzelne HTML-Steuerzeichen zu Fehlern bei der Generierung der Modelle führen würden. Die Darstellung der Texte im Emendo LMS als HTML stellt kein Problem dar, da diese Art der Darstellung be-



reits implementiert ist. Dies begründet sich dadurch, dass die Möglichkeiten zur HTML-basierten Formatierung von Texten im Designer bereits durch das Einfügen entsprechender Formatierungstags gegeben sind. Allerdings wird auf eine Formatierung mit Cascading Style Sheets (CSS) in dieser Phase des Prototyps verzichtet. Auf die Integration eines Vorschlagswesens für Lerninhalte (Mediensammlung) und häufig verwendete Regeln oder Kombinationen aus Modellelementen, wurde bereits eingegangen. Eine weitere sinnvolle Erweiterung stellt die Integration einer Vorschaufunktion in den Emendo Designer dar, welche beispielsweise die grafische Aufbereitung einer Single Choice-Aufgabe ihrer Bestandteile veranschaulicht. Diese Integration könnte entweder als Teil eines Tooltips erfolgen oder in Form einer Eclipse-View, die im Emendo Designer dauerhaft eingeblendet werden kann und eine Vorschau zu einem vom Benutzer markierten Modellelement bereitstellt. Weiterhin könnte das Einfügen bzw. Anhängen von Dateien (z. B. PDFs) an Lernbausteine unterstützt werden. Dies hätte den Vorteil, dass einzelne bestehende Lernmaterialien nicht nach Emendo adaptiert werden müssten und Lernende diese Dateien im LMS beispielsweise herunterladen könnten. Eine derartige Erweiterung hätte jedoch zur Folge, dass das Metamodell angepasst werden müsste und zwar dahingehend, dass Dateipfade zu Dateien als Attributwerte von Lernbausteinattributen eingefügt werden können. Diese Dateipfade könnten dann relativ die angehängten Dateien aus dem Modell heraus referenzieren und diese Referenzen im LMS schließlich aufgelöst werden. Die Interaktion mit dem Designer wird von mehr als der Hälfte aller Probanden als intuitiv beschrieben. Eine Einschränkung stellt jedoch dar, dass nicht die gesamte Zeichenfläche des Designers verwendet werden kann, um Modellelemente abzulegen. Demnach zielt diese Kritik auf die Unterscheidbarkeit von Gruppen, die Lernelemente gruppieren und Regeln ab. Diese Unterscheidung wurde jedoch bewusst getroffen, da Regeln auch dafür verwendet werden sollen, Gruppenzustände zu erfassen und zu überprüfen und ggf. Konsequenzen auszuführen. Um diese Funktionalität umzusetzen, ist eine Unterscheidbarkeit in Lernelemente und Regeln unabdingbar. Der Umgang mit dem Designer wird als schnell erlernbar beschrieben, sofern jedoch ein gewisses Maß an technischer Affinität bei Benutzern vorhanden ist. Als besondere Hürde in der Erlernbarkeit des Umgangs mit dem Designer ist die Definition von Bedingungen von Regeln durch ein Proband aus der Wirtschaftswissenschaft beschrieben. Diese Hürde sei für nicht-Informatiker nicht zu meistern, so Proband C. Diesem Argument kann dadurch entgegengetreten werden, dass andere Testpersonen, ebenfalls aus Informatik-fernen Fachbereichen, zwar eine Hürde in der Regeldefinition sahen, jedoch dies nicht als ausschlaggebenden

Grund gegen eine Verwendung anführen. Darüber hinaus könnte eine Änderung der Dialogstruktur in Verbindung mit einer Anpassung der Benennungen diese Hürde drastisch verkleinern.

#### 8.2.6 *Methodische Kritik und Grenzen*

Die vorgestellten Ergebnisse sind methodisch unter den folgenden Faktoren zu betrachten: Es handelt sich, bedingt durch die Forschungsfragen und damit die Ziele der Evaluation, um eine qualitative Erhebung, die aus einem vergleichsweise kleinen Sample besteht und damit zwar eine für die Forschungsarbeit wertvolle Validierung leistet, jedoch die quantitative Untermauerung der gewonnenen Ergebnisse nicht leistet. Konkret würde eine äquivalente quantitative Erhebung zu in der Evaluation untersuchten Forschungsfragen die gewonnenen Erkenntnisse verdichten und anreichern. Zudem könnte die Wichtigkeit von Thesen oder Aussagen zu Emendo nicht nur über eine Frequenzanalyse intendiert werden, sondern bei entsprechender Samplegröße, statistisch haltbar validiert und Thesen verifiziert oder falsifiziert werden.

Darüber hinaus ist eine positive Voreingenommenheit der Probanden in Bezug auf den Forscher durch ein persönliches Gespräch nicht auszuschließen. Damit könnten die Ergebnisse dahingehend aufgewertet worden sein, dass nicht das vollständige Bewertungsintervall durch Probanden ausgenutzt wurde. Dieser mögliche Umstand wurde dadurch abgefangen, dass vom Forscher zu Beginn jeder Evaluation geäußert wurde, dass Probanden sich in einer neutralen Haltung zum Untersuchungsgegenstand befinden, mit der Konsequenz, bewusst empfänglich auch für negative Aussagen zu sein, um Verbesserungspotentiale für Emendo identifizieren zu können. Dadurch konnten „[...] Bedingungen der Möglichkeit von Wahrhaftigkeit“ (Scheele u. Groeben, 2010, S. 507) geschaffen werden, in denen Probanden möglichst unverzerrt ihre Innensicht mitteilen können. Eine weitere Abhilfe in Bezug auf eine mögliche positive Voreingenommenheit der Probanden hätte womöglich die Tatsache geschaffen, dass eine andere Person die Befragung durchgeführt hätte. Damit wäre zwar dem Qualitätskriterium der Objektivität (Mayring, 2015, S. 124) Rechnung getragen, jedoch könnte angenommen werden, dass Fragen der Probanden an die interviewende Person in Bezug auf Emendo nicht fundiert beantwortet hätten werden können. Damit wäre ein verzerrtes Bild der Innensicht von Probanden entstanden, was auf einer unvollständigen oder fehlerhaften Information durch einen anderen Interviewer beruht hätte (Scheele u. Groeben, 2010, S. 507f.). Obwohl von Heinsen u. Vogt die Anwendung der SLT mit einem möglichen Auftreten eines Vollständigkeitszwangs beschrie-

ben wird, bei dem vorher nicht gesehene Zusammenhänge konstruiert und beschrieben werden (Heinsen u. Vogt, 2003, S. 108), so kann für die Durchführung festgehalten werden, dass dies zwar stellenweise auftrat, jedoch dabei immer als Qualitätskriterium die Zustimmung des Probanden durch den Interviewer eingeholt wurde. Letztere bezeichnen Scheele u. Groeben als „[...] oberstes Validitätskriterium für die Erfassung der Innensicht des Erkenntnisobjekts“ (Scheele u. Groeben, 2010, S. 507f.). Folglich wird ein möglicher Vollständigkeitszwang nicht als Problem gesehen, sondern vielmehr als Chance, neue und zuvor nicht angenommene Zusammenhänge oder Aussagen zu identifizieren, um daraus Verbesserungsmöglichkeiten für Emendo abzuleiten.

### 8.3 EVALUATION DER ANFORDERUNGEN AN DIE DSML

Nach der empirischen Validierung soll auch eine Evaluation der generischen Anforderungen an die erstellte DSML erfolgen und damit der Empfehlung von Frank gefolgt werden, der formuliert: „First, the evaluation of a DSML and a corresponding modeling tool recommends checking them against the requirements—both generic and specific [...]“ (Frank, 2013, S. 154).

Nach Frank kann durch die Validierung der Anforderungen der pragmatische Charakter und die Qualität einer DSML sichergestellt werden (Frank, 2013, S. 136f.). Deshalb soll im Folgenden bezugnehmend auf die in Kapitel 5.4.3 formulierten Anforderungen beschrieben werden, wie diese in der DSML umgesetzt wurden. Die Umsetzung der in Tabelle 15 konzeptuellen und damit spezifischen Anforderungen an die DSML, wurde bereits geleistet und soll an dieser Stelle nicht erneut betrachtet werden.

#### 8.3.1 *Semantische und grafische Affordance*

Die semantische und grafische Affordance beschreibt die semantisch eindeutige Verwendung der aus der Domäne stammenden Terminologie in der DSML. Die Einbettung in einer entsprechenden Notation soll nicht nur die Erlernbarkeit der DSML steigern, sondern auch zur intuitiven Anwendung beitragen. Um diese Anforderung zu erfüllen wurde zunächst eine Dokumentenanalyse durchgeführt und durch eine Lernplattformanalyse angereichert. Damit konnte die in der Domäne relevante Terminologie erfasst werden. Diese wurde als Grundlage für eine Nutzerbefragung mit Domänenexperten verwendet, die zum Ziel hatte, eine für die Emendo DSML einschlägige Notation zu entwickeln. Die empirische Evaluation zeigt, dass trotz dieser Vorarbeiten, Nachbesserungen in der Notation, vor allem in der Verwendung der Be-

griffligkeiten, vollzogen werden können. Dennoch kann diese Anforderung als erfüllt angesehen werden, da etwaige Änderungen der Terminologie der Emendo DSML rasch vollzogen werden können und darüber hinaus die Erlernbarkeit der Sprache sowie deren intuitive Verwendung mehrfach positiv in der Evaluation hervorgehoben wurden. Diese Äußerungen erfolgten auch von Probanden, die wie bereits aufgezeigt, aus nicht-technisch orientierten Fachbereichen stammen.

### 8.3.2 *Semantische Eindeutigkeit*

Die semantische Eindeutigkeit besagt, dass diese Anforderung dann umgesetzt ist, wenn die Konzepte der DSML in deren Anwendungsfokus genau eine semantische Bedeutung besitzen. Da diese Anforderung für alle Konzepte der DSML, vor allem bei der Erstellung des Emendo Designers in Sirius umgesetzt wurde, kann die generische Anforderung als erfüllt angesehen werden. Jedes Konzept des Metamodells der DSML wurde in Sirius genau einer Repräsentation zugeordnet und damit die semantische korrekte Verwendung für jedes Konzept Werkzeug-seitig sichergestellt. Unterstützt wird diese Aussage durch die Ergebnisse der Evaluation, die zeigen, dass die Sprachlogik und die Nachvollziehbarkeit der DSML positiv beurteilt werden und zu einem intuitiven und logischen Umgang mit der Sprache beitragen.

### 8.3.3 *Erweiterbarkeit*

Eine Erweiterbarkeit der Emendo DSML wurde dahingehend erreicht, dass konkrete Domänenkonzepte, wie beispielsweise das Konzept Single Choice-Aufgabe oder Textblock jeweils abstrakte Basisklassen besitzen. Darüber hinaus sind auf jedem weiteren darüber liegenden Abstraktionslevel nur abstrakte Klassen in dem Metamodell der DSML zu finden. Zulässige bzw. nicht zulässige Abhängigkeiten von Klassen sind mit AQL in Sirius definiert und nicht Teil des Metamodells. Damit genügt das Metamodell dem Offen-Geschlossen-Prinzip (Starke, 2014, S. 169f.), das besagt, dass Erweiterungen ohne Änderungen des bestehenden Quellcodes möglich sein müssen. Folglich kann diese Anforderung als erfüllt angesehen werden.

#### 8.3.4 *Separierte Abstraktionslevel*

Die Anforderung Abstraktionslevel in der Anwendung der DSML zu separieren ist dahingehend erfüllt, dass die Modellierung mit der Sprache zwischen Struktur- und Inhaltsebene unterscheidet. Die Struktur stellt das Metamodell der DSML, während die Inhaltsebene durch die Integration der DSML in den Designer geleistet wurde. Unter Berücksichtigung der Struktur, also auf Basis des Metamodells, werden im Designer Modellinstanzen der DSML durch Benutzer erzeugt, die über die Inhaltsebene und unter Nutzung von Dialogen, entsprechend mit Informationen angereichert werden. Die Evaluation zeigt, dass diese Separierung von Probanden positiv wahrgenommen wird und zu einem intuitiven Modellieren beiträgt, weshalb diese Anforderung als erfüllt gilt.

#### 8.3.5 *Eindeutigkeit von Sprachkonzept zu Zielkonzept*

Diese Anforderung definiert, dass die Emendo DSML alle Informationen in Modellen beinhalten soll, die für eine korrekte und eindeutige Umsetzung der Sprachkonzepte in Zielkonzepte benötigt werden. Zudem sollen Benutzer keinen Einfluss auf diese Beziehung zwischen Sprachkonzept und Zielkonzept haben, mit dem Ziel, semantische Variationen auszuschließen. Um diese Anforderung zu erfüllen, wurde für Benutzer nicht veränderbare Funktionalität in den Emendo Designer integriert, um Sprachkonzepte aus Modellinstanzen über einen Generator in Zielkonzepte umzuwandeln. Diese Umwandlung geschieht nach Anstoß durch Benutzer vollautomatisch und ohne deren Einfluss auf das Umwandlergebnis. Benutzer können lediglich Einfluss auf Rahmenbedingungen nehmen, wie den Zielpfad des Outputs des Generators. Die Logik sowie die benötigten Informationen für die Generierung von Modellinstanzen, sind demnach beide nicht Teil der Sprache. Folglich ist diese Anforderung erfüllt.

Zusammenfassend lässt sich festhalten, dass die generischen Anforderungen an die Modellierung mit der Emendo DSML alle, wie beschrieben, erfüllt wurden. Die Umsetzung dieser bezieht sich, wie aufgezeigt, nicht allein auf die DSML sondern tangiert ebenso deren Verarbeitung durch den Emendo Generator sowie deren Einbettung in den Emendo Designer.

#### 8.4 BEISPIELHAFTER EINSATZ FÜR SPIELIFIZIERTES JUST-IN-TIME TEACHING

Im Folgenden soll beispielhaft dargestellt werden, wie Emendo für die Umsetzung eines spielifizierten JiTT Lehr-Lernarrangements im Softwareengineering verwendet werden kann. Wie bereits aus den Ergebnissen der vorgestellten Evaluation hervorgeht, ist ein Einsatz von Emendo prinzipiell nicht nur auf Informatik-nahe Fachbereiche oder Teilfachbereiche der Informatik beschränkt. Dennoch soll bei diesem Anwendungsbeispiel der Fokus auf dem Fachbereich Softwareengineering liegen, bedingt durch den Anfertigungskontext dieser Arbeit, das BMBF-Projekt EVELIN (Abke et al., 2012a).

Zunächst soll als Lehr-Lernkonzept JiTT mit samt seinen Bestandteilen beschrieben werden. Darauf folgend wird das Konzept von spielifiziertem (gamified) JiTT<sup>10</sup> vorgestellt und bezugnehmend auf die in Kapitel 3.3.4 festgelegten Prinzipien für das Design von spielifizierten Lehr-Lernarrangements gezeigt, wie eine beispielhafte Umsetzung einer Lerneinheit zu Zustandsmuster mit Emendo erfolgen kann.

##### 8.4.1 *JiTT als Lehr-Lernkonzept*

Beim JiTT handelt es sich um ein Lehr-Lernkonzept, welches ursprünglich von Novak et al. für den Bereich der Physiklehre entwickelt wurde und seither in verschiedensten weiteren Bereichen der Lehre eingesetzt wird, z. B. Software Engineering (Hagel et al., 2013; Tao et al., 2014; Tao et al., 2015; Kamph et al., 2013), Datenbanken (Martinez, 2012), Einführung in die Informatik (Bailey u. Forbes, 2005), Programmieren (Carter, 2009; Carter, 2012; Astrachan, 2004; Gurka, 2012), Systeme und Design (Davis, 2009), Theoretische Informatik (Fleischer, 2004), Algorithmen und Datenstrukturen (Pargas, 2006). JiTT zielt darauf ab Lernaktivitäten, die innerhalb und außerhalb einer Veranstaltung stattfinden systematisch zu verbinden und greift dabei auf webbasierte Aufgaben zurück (Novak et al., 1999, S. 4; Novak, 2010, S. 5). JiTT wird als eine Instanz der aktiven Lehre und Möglichkeit eines Blended-Learning-Ansatzes verstanden (Waldherr u. Walter, 2009, S. 82). Novak et al. sprechen in diesem Zusammenhang von einer Unterteilung in *high-tech* und *low-tech* Elemente (Novak et al., 1999, S. XV). Während *high-tech* Elemente sich vornehmlich auf das Web beziehen, um die Kommunikation zwischen Lehrenden und Studierenden zu unterstützen und darüber auf das Curriculum abgestimmte Mate-


<sup>10</sup> Konzeptuell wurde der allgemeine Ansatz bereits ausführlich beschrieben und diskutiert in Bartel u. Hagel (2016a).

rialien ausgetauscht werden, fokussieren *low-tech* Elemente direkt die Lehre innerhalb der Veranstaltungen (Novak et al., 1999, S. XV). Durch Letztere wird eine Umgebung geschaffen, welche von Studierenden als „engaging and instructive“ (Novak et al., 1999, S. XV) wahrgenommen werden soll. Novak et al. sprechen Studierenden grundlegende Potentiale zu, die an einer Veranstaltung teilnehmen, welche JiTT als Lehr-Lernkonzept eingesetzt. Die Autoren sehen eine Verbesserung von überfachlichen Kompetenzen, vor allem solche, welche die Problemlösefähigkeit und die dafür notwendige Fähigkeit des Konzeptverständnisses sowie der kognitiven Vorstellung, betreffen (Novak et al., 1999, S. 4). Außerdem sollen durch JiTT soziale Fähigkeiten von Studierenden explizit angesprochen und verbessert werden. Hierzu zählen Novak et al. die Fähigkeit im Team zu kommunizieren und zu arbeiten (Novak et al., 1999, S. 4). Zudem wird die Praxisrelevanz des Ansatzes, sowie die Möglichkeit für Studierende, ihren Lernprozess eigens zu gestalten, als weitere Vorteile empfunden (Novak et al., 1999, S. 4).

#### 8.4.2 Ablaufbeschreibung

Um JiTT näher zu beleuchten, soll zunächst eine strukturelle Sicht eingenommen werden und die Abläufe sowie relevante Prozessobjekte beschrieben werden. Nach Definition der Lernziele, wird darauf basierend eine spielifizierte Lerneinheit mit dem Emendo Designer konzipiert, welche Studierenden zur Verfügung gestellt wird und außerhalb der Lehrveranstaltung zu bearbeiten ist. Bezugnehmend auf die Lerneinheit, welche im klassischen JiTT zu Beginn Leseaktivitäten der Lernenden erfordert, werden sogenannte JiTT-Fragen (Novak, 2010) erstellt. Diese besitzen eine vorgegebene Bearbeitungszeit und werden von Studierenden eigenverantwortlich bearbeitet. Studierende setzen sich selbstständig mit den Materialien auseinander und stellen bei Bedarf Verständnisrückfragen während der Bearbeitung im Emendo LMS. Spätestens kurz vor Beginn der darauffolgenden Lehrveranstaltung (meist 1 Tag vorher, siehe z. B. Hagel et al. (2013), Martinez (2012)), sollten die Antworten zu den JiTT-Fragen den Dozierenden vorliegen. Letztere unterziehen diese einem systematischen Review, mit dem Ziel, die folgende Lehrveranstaltung neu zu strukturieren bzw. bestehende Konzepte oder Inhalte bedarfsgerecht abzuwandeln. Zudem werden einschlägige Freitextantworten von Studierenden gesammelt. Dabei kann es sich um Antworten handeln, die sich entweder durch ein hohes Maß an Korrektheit auszeichnen oder schlichtweg Verständnisschwierigkeiten aufdecken, auf die in der Veranstaltung explizit eingegangen werden kann. Die Antworten von den Studierenden erfüllen also den

Zweck einer Planungsgrundlage als auch einer bedarfsgerechten Navigation durch die folgende Lehrveranstaltung. Die Abbildung 92 zeigt ein Beispiel für eine auf Basis einer Studierendenantwort erstellte Folie für eine Lehrveranstaltung, jedoch ohne ein Einsatz von Emendo. Die Beispielfolie<sup>11</sup> zeigt eine problematische<sup>12</sup> studentische Antwort auf die JiTT-Frage hin, wie in Java eine Klasse zur Laufzeit nachgeladen und instanziiert werden kann. Diese Fragestellung ist eingebettet in eine JiTT-Lehreinheit, bei der sich Studierende durch das Studieren von Lernmaterialien und das Bearbeiten von Aufgaben mit dem Plug-In Architekturstil (Starke, 2014, S. 277ff.) vertraut machen. Hierfür wurde den Studierenden – wie oben bereits beschrieben – ausgewähltes Lesematerial zur Verfügung gestellt und an den Lernzielen orientierte JiTT-Fragen gestellt nach (Figas et al., 2014), die qualitätssichernd für den Lernfortschritt der Studierenden sind, diesen Fortschritt für Lehrende transparent machen und von Studierenden in Eigenregie beantwortet wurden.



### Aufgabe 5 – Klassen zur Laufzeit nachladen und instanziiieren

Die Methode lädt eine Klasse mit dem Namen *IchBinEinKlassenname* dynamisch zur Laufzeit in die JVM, d.h. *IchBinEinKlassenname* wird initialisiert (alle static-Blöcke darin ausgeführt).

**ACHTUNG!** Beim Ausführen der Methode wird kein Objekt von *IchBinEinKlassenname* erzeugt!

Beispielcode  
(problematische  
Antwort):

```
package de.beispielReflection
public class IchBinEinKlassenname {
    static {
        System.out.println("static block in IchBinEinKlassenname");
    }
}

public class Program {
    public static void main(String[] args) {
        try {
            Class c = Class.forName("de.beispielReflection.IchBinEinKlassenname");
        } catch (ClassNotFoundException e) {
            System.out.println("Klasse mit Namen IchBinEinKlassenname nicht gefunden.");
        }
    }
}
```

Instanziiieren mit:  
 IchBinEinKlassenname obj = (IchBinEinKlassenname) c.newInstance();

37 JiTT Plug-in, Softwarearchitektur; Alexander Bartel und Georg Hagel

Abbildung 92: Beispielfolie einer JiTT-Lehrveranstaltung

Richtige oder gute Antworten zu einer Frage werden ebenso bewusst aufgegriffen und angesprochen wie problematische Antworten. Dies ist dem Umstand geschuldet, dass für ein erfolgreichen Lernen (in diesem Fall asynchrones) Feedback – ob gut oder schlecht – ein wichtiger Bestandteil ist (Novak,

- 11 Beispielhafter Auszug aus der Lehrveranstaltung Software Architektur vom Sommersemester 2015 an der Hochschule Kempten.
- 12 Problematisch deshalb, da die Antwort lediglich das Nachladen einer Klasse beinhaltet und nicht den nächsten geforderten Schritt der Instanziierung.



2010, S. 6). In der Lehrveranstaltung selbst werden aufbauend auf den Antworten der Studierenden durch die Lehrenden Möglichkeiten geschaffen, in Form von kleinen Gruppendiskussionen (Hagel et al., 2013), Demonstrationen/Präsentationen (Novak et al., 1999, S. 39), dem Einsatz von Klickern (Lasry, 2008) oder FlashCards (Lasry, 2008) oder kooperative Lernaktivitäten (Crouch et al., 2007) das Verständnis für Studierende weiter zu steigern (Novak, 2010, S. 6f.).

#### 8.4.2.1 *Aufgaben beim JiTT*

Eines der zentralen Elemente bei diesem Lehr-Lernansatz sind die Aufgaben. Die Macher von JiTT messen diesen eine besondere Bedeutung zu, da sie nicht nur zum Lernen der Studierenden eingesetzt werden, sondern eine Reihe weiterer Ziele verfolgen, wie z. B. die Transparenz auf den Lernfortschritt der Studierenden darzustellen oder dabei helfen, potentielle Verständnisprobleme seitens der Studierenden aufzudecken (Novak, 2010, S. 11). Auch tragen sie nach Riegler dazu bei, formative Assessments zu ermöglichen, in dem Aufgaben – sofern möglich – automatisch bewertet werden können (Riegler, 2012, S. 89). Novak et al. unterscheiden prinzipiell zwei Kategorien von JiTT-Aufgaben gemäß ihrem Einsatzzweck, die sogenannten WarmUp- und Puzzleaufgaben (Novak et al., 1999, S. 20).

#### 8.4.2.2 *WarmUp-Aufgaben*

Die WarmUp-Aufgaben werden als Herz des JiTT-Ansatzes bezeichnet, da sie eine Feedbackschleife zwischen solchen Lernaktivitäten bilden, die während und außerhalb einer Lehrveranstaltung stattfinden (Novak, 2010, S. 6; Novak et al., 1999, S. 21). Sie dienen vornehmlich dazu, evtl. bereits vorhandene Vorannahmen zu einem Thema zu relativieren und bei Bedarf neu zu überdenken oder unvollständige Vorannahmen zu vervollständigen (Novak et al., 1999, S. 20). Für diesen Prozess wird von Novak et al. unterstellt, dass in Kombination mit einem Realweltbezug Themen Sachverhalte durch Lernende besser verstanden und verinnerlicht werden (Novak et al., 1999, S. 20). Nach Cashman u. Eschenbach sowie Martinez bestehen WarmUp-Aufgaben aus zwei bis drei Aufgaben, die mit einer kurzen Antwort beantwortet werden können (Cashman u. Eschenbach, 2003; Martinez, 2012). Hinzu kommt eine Feedback Box, in der Lernende weitere Fragestellungen angeben können, die nicht von den WarmUp-Aufgaben abgedeckt werden. Auch Single bzw. Multiple Choice-Aufgaben stellen mögliche Aufgabentypen dar (Martinez, 2012; Novak et al., 1999). In der Einführungsveranstaltung aus einem Ingenieurstudiengang, in dem Cashman u. Eschenbach JiTT einsetzten, nutzten sie eine Aufgabe, in

der die Studierenden eine kurze Berechnung durchführen sollten. Für eine weitere Aufgabe (Konzeptfrage<sup>13</sup>, engl. ConcepTest), wurde eine offene Antwort vorgesehen, in der Studierende ein bestimmtes Konzept rekapitulieren und die Aufgabe durch eine einschlägigen Niederschrift beantworten sollten (Cashman u. Eschenbach, 2003). Für die Bewertung dieser Antwort flossen sowohl die Qualität des niedergeschriebenen Gedankenguts als auch die Detaillierung der Informationen als Bewertungskriterien mit ein. So konnten Studierende die maximal mögliche Punktzahl bekommen, auch wenn die Antwort auf die Frage inkorrekt war, jedoch das Vorgehen und zu Grunde gelegte Annahmen ausreichend gut dokumentiert waren (Cashman u. Eschenbach, 2003). Anders ausgestaltet finden sich WarmUp-Aufgaben bei Kamph et al. (2013). Sie nutzen ein zweiteiliges Quiz im Kontext einer Software Engineering Vorlesung als WarmUp-Aufgabe, bei dem der erste Teil aus einer Einleitung besteht und der zweite Teil den Fragenteil bildet (Kamph et al., 2013, S. 11f.). Auf die im Fragenteil enthaltenen Aufgaben<sup>14</sup> wird nicht näher eingegangen, jedoch merken die Autoren an, dass „[...] es nicht darauf [ankommt], die gestellte Frage exakt zu beantworten“ (Kamph et al., 2013, S. 12). Vielmehr stehe im Fokus, den Studierenden die Komplexität eines Softwaresystems zu verdeutlichen und reflexiv sowie vorbereitend mit Hilfe der Aufgaben, Gedanken zu der anstehenden Lehrveranstaltung zu machen (Kamph et al., 2013). Diese Kriterien führen dazu, dass eine Bewertung der einzelnen Antworten zu den Aufgaben nur auf Basis dessen erfolgt, ob die bewertende Person den Eindruck erhält, dass der jeweilige Lernende sich ausreichend mit der Fragestellung auseinander gesetzt hat (Kamph et al., 2013, S. 12; Böttcher et al., 2015, S. 454). Dieser Einsatz von Aufgaben in der WarmUp-Phase impliziert also nicht in erster Linie die Erfüllung eines Leistungsziels, sondern vielmehr den Zweck eines Lernziels. Somit wird beim Bearbeiten einer Aufgabe schon an der Aufgabe selbst gelernt, wobei die Aufgabe den Lernprozess initiiert, steuert und organisiert (Seel, 1981, S. 14; Figas et al., 2014).


#### 8.4.2.3 *Puzzleaufgaben*

Die zweite Kategorie beim JiTT bilden die von Novak et al. als Puzzle bezeichneten Aufgaben (siehe Abbildung 93). Bei dieser Form von Aufgaben, welche typischerweise nur eine Frage beinhalten, sollen sich die Lernenden mit Themen auseinandersetzen, die aufgrund der Vielfalt an eingesetzten Konzep-

13 Beispielhaft könnte man als Repräsentant einer derartigen Frage formulieren: Was hat das Zustandsmuster mit dem Offen-Geschlossen-Prinzip zu tun?

14 Die Autoren geben nur einen zeitlichen Horizont für die Bearbeitung aller Aufgaben von 30 Minuten an, um den Aufwand für die Studierenden begrenzt zu halten.

ten, eine höhere Komplexität besitzen (Novak et al., 1999, S. 22). Die Ausgabe von Puzzles erfolgt im zeitlichen Turnus der WarmUp-Aufgaben. Allerdings eignen sich Puzzles vornehmlich dazu, ein bereits behandeltes Thema zu rekapitulieren und werden damit als ein abschließendes didaktisches Element angesehen (Novak et al., 1999).



**IUPUI**  
Department of Physics

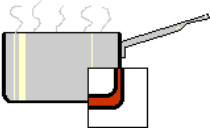
**This Week's  
Puzzle**

---

- Puzzle solutions are due by 9:00 AM on Friday of each week.
- Puzzles are worth 5 points of extra credit each.
- Answers must be accompanied by valid reasoning. Just like the tests, the answer alone isn't enough!
- Each week, the first student to submit a correct answer will receive the stylish [251 T-shirt](#).
- Previous winners get a "virtual t-shirt." No shirt, but a chance to win a prize with actual value at the end of the semester (often a book, or maybe software).
- Please enter your solution in the text area at the bottom of this page. DON'T FORGET TO GIVE YOUR NAME!

## "Pot Physics"

Due 9:00 AM Friday, January 15, 1999



Advertisements for some expensive cookware have diagrams that look much like the figure above. The diagram is labeled to indicate that the inner and outer surfaces of the pot are made of stainless steel, and the inner layer is made of another metal such as aluminum or copper.

Please explain why pots are made this way. Discuss why the different metals are chosen, and why a layered structure is better than using one thick layer of one of the metals.

Don't forget to explain your answer as clearly as possible.

Type your answer into the box and click the submit button.  
The time and date of your entry will be recorded.

Abbildung 93: Beispiel eines JiTT Puzzles (Patterson, 1999)

Die Macher von JiTT weisen darauf hin, dass Ergänzungsmaterial wie beispielsweise Fachartikel oder Weblinks dazu beitragen, die Verbindung zur Realwelt herzustellen und motivationsförderlich auf Studierende wirken können (Novak et al., 1999) sowie den Nutzen des erworbenen Wissens aufzeigen – eine These, die auch von anderen Forschern angeführt wird (Figas et al.,

2013). Das Lesen und Auseinandersetzen mit Ergänzungsmaterial<sup>15</sup> sollte als zusätzlichen Aufwand mit in die Erwartungshaltung der Lehrenden an die Lernenden einfließen und sich durch einen zusätzlichen Lernanreiz, beispielsweise in Form von Zusatzpunkten, auszeichnen (Novak et al., 1999).

#### 8.4.3 *Spielifiziertes JiTT*

Beim spielifizierten JiTT kommen Prinzipien der Spielifizierung, wie in Kapitel 3.3.4 beschrieben, im JiTT zum Einsatz. Die Motivation der Integration der Prinzipien in den Ansatz liegt in der Analyse von einschlägiger Literatur (Camp et al., 2010; Hagel et al., 2013; Martinez, 2012; Carter, 2012; Watkins u. Mazur, 2010), welche Anwendungen von JiTT und dabei aufgetretene Probleme beschreibt (Bartel u. Hagel, 2016a, S. 4-6). Es zeigt sich, dass Probleme beim Einsatz dieses Lehr-Lernkonzepts in mehreren Anwendungen unabhängig vom Anwendungskontext wiederfinden. Diese berichteten Probleme lassen sich zu vier Problemkategorien zusammenfassen, die auf motivationale Ursachen zurückzuführen sind und im Folgenden mit Designprinzipien aus Kapitel 3.3.4 als Lösungsvorschläge<sup>16</sup> beschrieben werden (Bartel u. Hagel, 2016a, S. 5f.):

- **Arbeit mit Lernmaterialien:** Lehrende berichten, dass Studierende die nötigen Vorarbeiten am Lernmaterial nicht wie gewünscht vollziehen, und führen dies auf ein Mangel an Interesse, fehlendes Verständnis oder eine zu hohe extrinsische Motivation zurück, welche nach mehrmaligem Einsatz abflachte.  
→ Einbettung von SMART-Zielen entlang individueller Lernpfade, welche auf Realwelterzählungen basieren, damit Studierende sich mit den Inhalten besser identifizieren können und die eine sukzessive Zielerreichung ermöglichen in Verbindung mit der Freiheit Fehler dabei zu machen.
- **Beantwortung von Fragen:** Die Beantwortungsquote von Fragen, die innerhalb einer JiTT-Einheit gestellt werden, ist gering, sofern deren Beantwortung nicht an eine Belohnung oder einen Benefit geknüpft ist.  
→ Anwendung von SMART-Zielen als ein Aspekt der transparenten

<sup>15</sup> Diese Typen von Aufgaben werden oftmals in der Literatur salopp „Good Fors“ genannt (Novak et al., 1999).

<sup>16</sup> Die Lösungsvorschläge basieren auf den Ausführungen in Bartel u. Hagel (2016a). Sie erheben nicht den Anspruch an Vollständigkeit, sondern stellen Möglichkeiten dar, die sich durch einschlägige Literatur in Kapitel 3 untermauert, als erfolgsversprechend in (teilweise in ähnlichen Problemkontexten) herausgestellt haben.

Zielorientierung, stärkere soziale Einbindung von Individuen, Schaffung von Möglichkeiten des sozialen Vergleichs beim Lern- bzw. Bearbeitungsfortschritt.

- **Qualität der Antworten:** Die Qualität der Antworten liegt unter den Erwartungen, die seitens der Lehrenden bestehen. In einzelnen Fällen wird das Plagiiere von Antworten berichtet, um das Erreichen von Lernzielen vorzutäuschen.  
→ Integration von anreizgestützter Fortschrittsüberwachung mit für Mitlernende sichtbaren Errungenschaften, kontextbezogenes und nicht verallgemeinertes Feedback sowie Integration von frei wählbaren Lernpfaden, die keine Bearbeitungsreihenfolge vorgeben, um das Autonomieempfinden zu steigern und dabei vermehrt auf Aufgaben setzen, die zum individuellen Nachdenken anregen und dies im Aufgabentyp verankern (z. B. Freitextaufgaben in Emendo bzw. Puzzle-Aufgaben im klassischen JiTT).
- **Fehlende Transparenz:** Die fehlende Transparenz bezogen auf den Ablauf einer JiTT-Einheit führt zusammen mit einem nicht-Anknüpfen an den persönlichen Lernprozess zu Ablehnung, bedingt durch Unsicherheit und Skepsis.  
→ Individuelle Lernpfade, die eine individuelle Schwierigkeit berücksichtigen und freie Bearbeitung der Lernbausteine zulassen, kontextbezogenes und individuelles Feedback in Verbindung mit der Freiheit Fehler bei der Bearbeitung zu machen.

Die vorgestellten Lösungsmöglichkeiten lassen sich nun weiter operationalisieren, z. B. in dem Mechanics oder Components für deren Umsetzung in Emendo festgelegt werden. Aus Funktionssicht lassen sich alle oben genannten Verbesserungsvorschläge zu besagten Problemen in Emendo abbilden, sowohl im Emendo Designer bei der Konzeption einer spielifizierten JiTT-Lerneinheit als auch bei der Bearbeitung dieser im Emendo LMS durch Lernende.

#### 8.4.4 *Beispielhafte Umsetzung einer spielifizierten JiTT-Lerneinheit*

Beispielhaft<sup>17</sup> soll im Folgenden gezeigt werden, wie eine spielifizierte JiTT-Lerneinheit unter Berücksichtigung der vorgestellten Lösungsmöglichkeiten

<sup>17</sup> Die folgenden Ausführungen basieren auf der bereits publizierten Lerneinheit zu finden in Bartel u. Hagel (2016a, S. 10ff.).

im Emendo Designer modellhaft für das Thema UML-Zustandsdiagramme dargestellt werden kann. An dieser Stelle kann der GCDP bzw. der GCDR hilfreich sein, um Hilfestellungen beim generellen Vorgehen zu leisten oder Unterstützung bei der sukzessiven Konzeption einer derartigen Einheit zu bieten, bevor diese im Designer modelliert wird.

Für die Lerneinheit werden gemäß GCDP zunächst Lernziele definiert. Dabei wird angenommen, dass Lernende bereits mit vorgelagerten Techniken im Softwareprozess, wie der Anforderungserhebung und dem Modellieren von UML-Zustandsdiagrammen vertraut sind. Nach der Absolvierung der spielifizierten JiTT-Lerneinheit, sind Studierende in der Lage

- grundlegende Konzepte des Zustandsmusters zu verstehen,
- diese zu beschreiben,
- das Zustandsmuster für neue Szenarien in Form eines Klassendiagramms und Code anzuwenden und
- existierende Zustandsmuster zu beurteilen.

Methodisch stützt sich die beschriebene Lerneinheit auf den bereits vorgestellten Ansatz des spielifizierten JiTT und integriert daher beispielhaft spielifizierte Elemente, die im vorherigen Kapitel aus den prinzipiellen Lösungsvorschlägen zu existierenden JiTT-Einheiten abgeleitet bzw. operationalisiert werden. Die Abbildung 94 zeigt ein Modell des Emendo Designers, in dem die Lerneinheit modelliert ist. Die einzelnen Bestandteile (Modellelemente) der Lerneinheit sollen im Folgenden näher detailliert werden. Dabei sei erwähnt, dass die Bezeichnung der Modellelemente verändert werden kann, um eine bessere Orientierung im Modell zu ermöglichen. Aufgrund des vergleichsweise geringen Umfangs der Lerneinheit wurde in diesem Fall darauf verzichtet. Grundsätzlich besteht die Lerneinheit aus insgesamt 4 Teilen: Einer motivierenden Erzählung (1), WarmUp-Aufgaben, um an das vorhandene Wissen anzuknüpfen (2), einem Lernteil in dem Lernende sich neues Wissen aneignen und dies anwenden (3) und einem Abschluss, welche Puzzle-Aufgaben enthält (4).

Der erste Teil der Lerneinheit umfasst einen Textblock (*Textblock\_6602*), welcher die Lerneinheit thematisch in eine Erzählung einbettet und im ersten Lernschritt motivierend für Studierende wirken soll. Darin könnte beispielsweise folgende Ausführung<sup>18</sup> zu lesen sein:

<sup>18</sup> Diese Erzählung wurde in englischer Sprache bereits veröffentlicht in Bartel u. Hagel (2016a) für eine konzeptuelle Ausarbeitung einer Lerneinheit, die das Lernen von UML-Zustandsdiagrammen fokussiert. Vorliegende Erzählung wurde inhaltlich auf Zustandsmuster adaptiert.

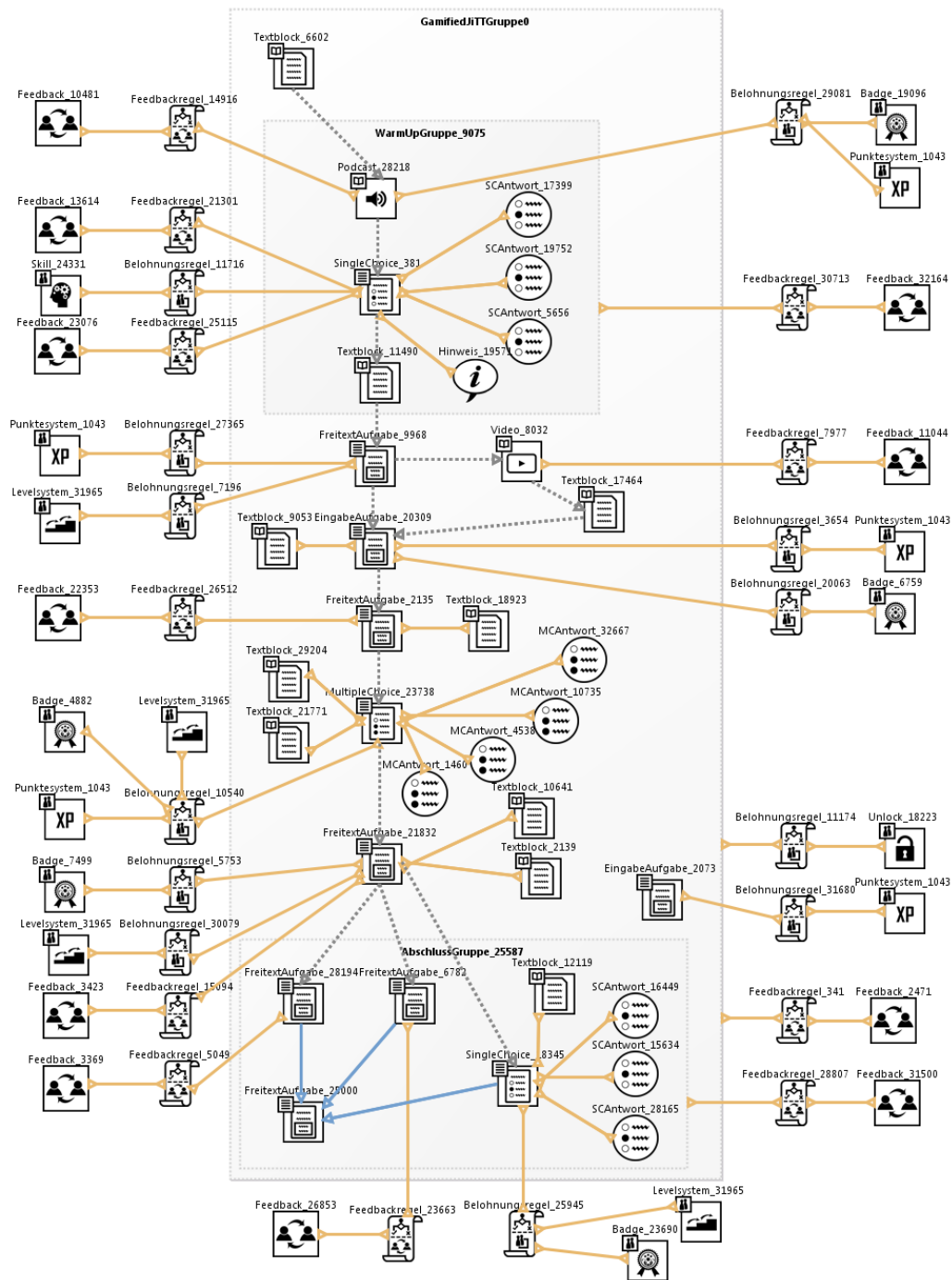


Abbildung 94: Modellbeispiel zu einer spielifizierten JITT-Einheit zum Thema Zustandsmuster

*Dein Name ist Alice, 27 Jahre alt und Du liebst Musik über alles. Seitdem du 12 Jahre alt bist, willst du dir einen Traum erfüllen und einen eigenen Musikclub führen. Und schließlich hast du es geschafft, du hast dir den Traum erfüllt! Aber... es läuft nicht wie erwartet. Als neue Chefin brauchst Du etwas, das Dein Clubkonzept von den Bestehenden abgrenzt, da Du dich in finanziellen Schwierigkeiten befindest.*

rigkeiten befindest. Die Themenpartys sind ein guter Anfang, aber trotzdem sind die Lichteffekte im Club langweilig und passen nicht zu den Themen der Partys – das hast Du nun schon mehrfach von Gästen Deines Clubs gehört. Ein Freund von dir, Mike, arbeitet im Lichtgeschäft und Du bittest ihn um einen Rat verschiedene Lichteffekte in den Club zu integrieren. Er schlägt eine Laserbar vor (siehe Abbildung 95), da dies das neueste und leistungsfähigste Konzept auf dem Markt ist.

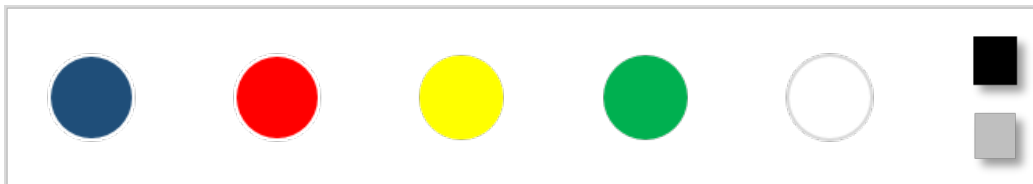


Abbildung 95: Laserbar aus initialen Erzählung der spielifizierten JiTT-Einheit

*Er sagt dir sogar, dass es programmierbar ist aber Mike hat keine Ahnung, wie das genau funktioniert. Er hat aber gelesen, dass es auf dem Zustandsmuster basiert und dass es wohl nicht schwer ist, dieses auch zu programmieren. Du kennst zwar Zustandsdiagramme, hast aber noch nie etwas vom Zustandsmuster gehört aber willst die Bar bis morgen Abend einsatzbereit haben, da dann die 90er-Jahre-Party stattfinden wird von der das weitere Bestehen des Musikclubs abhängig ist...*

Der darauffolgende Teil der WarmUp-Aufgaben beginnt mit einem Podcast (*Podcast\_28218*) in dem Lernende sich einen Dialog<sup>19</sup> zwischen zwei Softwarearchitekten anhören, die über den Sinn und die Nützlichkeit von Mustern beim Design diskutieren. Dies erlaubt Lernenden eine allgemein Verortung des Themas. Sobald der Podcast vollständig angehört ist, wird über die angehängte Feedbackregel (*Feedbackregel\_14916*) das Gelernte reflektiert und Studierende mit einem Badge „The patient listener“ und Erfahrungspunkten belohnt. In der darauffolgenden Single Choice-Aufgabe (*SingleChoice\_381*) beantworten Lernende eine Frage, die sich auf den vorangegangenen Podcast bezieht. Dafür können Sie eine Hilfestellung in Form eines Hinweis (*Hinweis\_19571*) annehmen. Je nach Richtigkeit der Antwort erhalten Sie entweder Feedback (*Feedbackregel\_21301*) bei einer falschen Beantwortung oder einen Skill (*Belohnungsregel\_11716*), bei einer richtigen Antwort. Sollte die Antwort falsch sein, sie wiederholen die Aufgabe und beantworten die Aufgabe richtig, so erhalten sie hierzu ebenfalls Feedback (*Feedbackregel\_25115*). Als letzten Teil der WarmUp-Aufgaben ist ein Text (*Textblock\_11490*), welcher das Gelern-

<sup>19</sup> Eignen würde sich hier beispielsweise der Heise Developer *SoftwareArchitektTOUR-Podcast* über den Einsatz von Patterns (Stal u. Völter, 2704).



te zusammenfasst und einen Ausblick auf die voraus liegenden Themen gibt, die es für Alice noch zu bearbeiten gilt. Ein entsprechendes Feedback (*Feedbackregel\_30713*) informiert über den erfolgreichen Abschluss der WarmUp-Aufgaben bzw. -lernbausteine. Als erster Baustein des Lernteils stellt die Freitextaufgabe (*FreitextAufgabe\_9968*) die erste Konfrontation mit dem neuen Zustandsmuster für Lernende dar. Die Aufgabenstellung präsentiert das UML-Klassendiagramms und fordert die Studierenden auf das Zustandsmuster mit dem bereits bekannten UML-Zustandsdiagramm im Hinblick auf die Laserbar zu vergleichen. Kriterien für die Bewertung von Antworten sind die fachliche Richtigkeit, Vollständigkeit und Ausdrucksfähigkeit. Je nach Erfüllungsgrad erhalten die Lernende weitere Punkte (*Belohnungsregel\_27365*) oder erreichen ihren ersten Levelaufstieg (*Belohnungsregel\_7196*), wobei letzterer an das erste Lernziel der Lerneinheit geknüpft ist. Im Anschluss daran können Studierende sich entweder die Funktionsweise der Laserbar in einem von Dozierenden selbst produzierten Video ansehen dafür Anerkennung in Form eines Feedbacks erhalten (*Feedbackregel\_7977*), anschließend einen Text (*Textblock\_17464*) lesen, welcher analoge Verwendungen des Musters für andere Anwendungskontexte vorstellt, beispielsweise für Ampeln, um anschließend mit der Beantwortung einer Eingabeaufgabe (*EingabeAufgabe\_20309*) fortzufahren. Darin werden Studierende gebeten die Anzahl an Zuständen einzugeben, die für die Laserbar zutrifft und die Erzählung weiterführt: *Als Du dir die Bar genauer ansiehst, entdeckst Du einen schwarzen Schalter, der die Bar anschaltet und der blaue Laser leuchtet. Der graue Schalter befindet sich direkt unter dem schwarzen Schalter. Du kannst ihn drücken, um den zweiten Laser zu beleuchten, den Roten, wobei der Blaue ausgeschaltet wird.* Ist die eingegebene Anzahl korrekt, erhalten Lernende einen Badge (*Belohnungsregel\_20063*). Ist sie inkorrekt und bedarf es einer Wiederholung der Aufgabe, werden Studierende mit Erfahrungspunkten belohnt (*Belohnungsregel\_1043*). Es folgt eine Freitextaufgabe (*FreitextAufgabe\_2135*) bei der Studierende die Zustände der Laserbar, abhängig von Objekten, konkret benennen und in Pseudocode die Schaltungsreihenfolge der einzelnen Zustände gemäß des Zustandsmusters beschreiben sollen. Die Bewertung der Abgaben erfolgt anhand der Kriterien richtige Anwendung des Musters, Vollständigkeit sowie richtiger Einsatz von Pseudocode. Bei Erfüllung der Kriterien, erhalten die Studierenden Feedback (*Feedbackregel\_26512*). Im Anschluss daran bewerten Studierende Aussagen zum Zustandsmuster mit Hilfe einer Multiple Choice-Aufgabe (*MultipleChoice\_23738*), wofür sie mit einem weiteren Levelaufstieg (gebunden an zweites Lernziel), einem Badge sowie Erfahrungspunkten belohnt werden, sofern die Aufgabe korrekt beantwortet wurde. Es folgt in der anschließenden

Freitextaufgabe (*FreitextAufgabe\_28132*), die Aufforderung das Zustandsmuster für die Laserbar selbstständig zu implementieren: *Du hast dir nun ausreichend Wissen angeeignet und Vorüberlegungen angestellt, um die Laserbar in Quellcode umzusetzen. Du willst ja schließlich nicht den ganzen Abend neben der Laserbar sitzen und nach einer gewissen Zeit den grauen Knopf drücken, oder?* Studierende geben also bei dieser Aufgabe Quellcode ab, welcher anhand der Kriterien richtige Anwendung des Musters, Vollständigkeit sowie richtiger Einsatz von Quellcode bewertet wird. Bei Erfüllung dieser Kriterien erhalten Studierende einen weiteren Levelaufstieg (*Belohnungsregel\_30079*), welcher ebenfalls an das dritte Lernziel geknüpft ist sowie einen Badge (*Belohnungsregel\_5753*) und Feedback (*Feedbackregel\_15094*) zur Rekapitulation des bisherigen Lernfortschritts.

Im vierten und letzten Teil der Lerneinheit folgen drei Auswahlmöglichkeiten, um im Lernpfad fortzufahren. In der ersten Freitextaufgabe (*FreitextAufgabe\_28194*), haben Studierende die Möglichkeit ihren Quellcode aus der vorherigen Aufgabe um eine zeitliche Verzögerung zu erweitern, damit die Zustände zeitlich (individuell) getaktet durchlaufen werden können. Sie erhalten Feedback (*Feedbackregel\_5049*), wenn sie diese Aufgabe beginnen. Bei der zweiten Auswahlmöglichkeit mit der Bearbeitung von einer weiteren Freitextaufgabe (*FreitextAufgabe\_6782*) können sie die bestehende Laserbar um eine weitere, darunterliegende Laserbar mit nur zwei Lasern nebenläufig erweitern. Auch hierzu erhalten sie Feedback (*Feedbackregel\_23663*), wenn die Aufgabe begonnen wurde. Die dritte Auswahlmöglichkeit besteht in der Beurteilung eines Alltags-Szenarios für Alice (dargestellt als UML-Klassendiagramm) aus der Sicht des Zustandsmusters, in dem ein Ampelsystem bestehend aus zwei Hauptstraßenampeln und einer Fußgängerampel beschrieben wird. Mit der korrekten Beantwortung steigen Lernende ein weiteres Level (*Belohnungsregel\_25945*) auf und erfüllen damit das vierte und letzte Lernziel. Zudem erhalten sie einen Badge. Im Anschluss folgt eine abschließende Freitextaufgabe (*FreitextAufgabe\_25000*), bei deren Bearbeitung Studierende die Möglichkeit haben, Dinge anzusprechen, die bisher entweder nach wie vor unklar im Bezug die Lerneinheit sind oder weitere Fragen zur Lerneinheit stellen oder Feedback dazu geben können. Dies ist aus dem klassischen JiTT stammend, ein übliches Vorgehen.

Werden alle Aufgaben aus der AbschlussGruppe\_25587 korrekt beantwortet wird Lernenden ebenfalls ein Skill verliehen (*Belohnungsregel\_12020*). Sind außerdem mehr als 2/3 aller Aufgaben in der gesamten Lerneinheit korrekt auf Anhieb beantwortet, wird durch die *Belohnungsregel\_11174* ein Unlock freigeschalten. Studierende haben damit die Möglichkeit als Zusatzaufgabe

die *EingabeAufgabe\_2073* zu lösen, wofür sie abschließend weitere Punkte bekommen (*Belohnungsregel\_31680*), wenn sie korrekte Antwort auf die Frage geben, wie das Muster heißt, mit dem das Eintreten in bzw. Austreten aus einem Zustand anderen Klassen bekannt gemacht werden kann. Mit *Feedbackregel\_341* wird die Lerneinheit dahingehend abgeschlossen, dass bei einer Bearbeitungsquote von mehr als 90% die 90-er Party von Alice ein Erfolg war und der Club gerettet werden konnte.

Das beschriebene Modell zeigt eine Möglichkeit, wie eine spielifizierte JiTT-Lerneinheit ausgestaltet werden kann. Da Emendo zwischen struktureller und inhaltlicher Spielifizierung unterscheidet, können beide Teilbereiche beliebig kombiniert und variiert werden. Die Bearbeitungsreihenfolge ist in der beschriebenen Einheit nicht vorgegeben. Dennoch sorgt das in das Emendo LMS integrierte und Strategie-basierte Vorschlagswesen für eine strukturierte Bearbeitungsreihenfolge der Lernbausteine, wobei Lernende von diesen Vorschlägen abweichen können. Ebenso wie die Inhalte und die Struktur der Lernbausteine innerhalb einer Lerneinheit kann deren Zweck bzw. Einbettung in den Lehrkontext variieren. Mit dem vorgestellten Modell ist zwar ein Blended-Learning-Ansatz beschrieben. Jedoch sind Anwendungen zum Selbststudium (z. B. als Vokabeltrainer für UML-Notationselemente) oder gezielt für eine Gruppenarbeit (z. B. wird innerhalb einer Lehrveranstaltung gruppenweise möglichst komplizierter Code geschrieben, dessen Funktionsweise von anderen Gruppen in einem gewissen Zeitraum korrekt wiedergegeben werden muss), ebenso wie Leistungsaufgaben zur Prüfungsvorbereitung in Eigenregie, ebenfalls denkbar und problemlos umsetzbar. Der mögliche Bearbeitungszeitraum einer Lerneinheit skaliert ebenso wie der Einsatzzweck, beispielsweise über mehrere Semester, auf ein Semester oder eine Einheit darin begrenzt, bis hin zu einem Teil einer Lehrveranstaltung (z. B. in Form eines One Minute Papers nach Waldherr u. Walter (2009, S. 55ff.)).

Das beschriebene Modell genügt zudem den Designprinzipien zum Design von spielifizierten Lehr-Lernarrangements, da es SMART-(Lern-)Ziele in Lernpfade einbettet, die für Lernende realitätsnah und transparent sind und deren Erreichung nicht an eine Reihenfolge gebunden ist. Zudem können im Emendo LMS bei einer entsprechenden Generierung des Modells, die Fortschrittsstände zwischen Lernenden verglichen werden, beispielsweise anhand der erspielten Badges, des Erfahrungspunktestandes oder der erreichten Level. Dies schafft Möglichkeiten des sozialen Vergleichs. Zudem kommen überwiegend Freitextaufgaben zum Einsatz, die durch Anreize gestützt, das Plagieren von Antworten erschweren. Darüber hinaus bietet sich durch die Funktionalität des Emendo LMS die Möglichkeit, die Antworten von Lernenden zu diesem

Fragetyp in der Lehrveranstaltung aufzugreifen und darauf einzugehen. Neben den Möglichkeiten des individuellen Feedbacks, die Emendo bietet, stellt dies eine weitere Möglichkeit dar, kontextbezogenes Feedback in den Lernprozess von Studierenden gezielt zu etablieren.

## 8.5 ZWISCHENFAZIT

Dieses Kapitel beschrieb eine Evaluation in Form einer Befragung von potentiellen Nutzern, bei der die Emendo DSML und der Emendo Designer im Vordergrund standen. Neben dem allgemeinen Konzept, welches durch die Teilnehmenden an der Evaluation beurteilt wurde, wurden ebenso Erfahrungen bei der Modellierung mit der Emendo DSML, eingebettet in den Emendo Designer geäußert. Die Ergebnisse zeigen, dass potentielle Nutzer Emendo überwiegend positiv sehen sich jedoch auch Verbesserungspotentiale gezeigt haben, die bei der Weiterentwicklung von Emendo beachtet werden sollten. Zudem wurden die in Kapitel 5.4.3 beschriebenen Anforderungen an die DSML mit der tatsächlichen Umsetzung der Sprache abgeglichen. Es zeigte sich, dass alle beschriebenen Anforderungen umgesetzt werden konnten. Dieses Kapitel zeigte zudem beispielhaft wie Emendo für das Blended-Learning-Konzept spielifiziertes JiTT genutzt werden kann. Dies wurde anhand eines Beispiels aus dem Software Engineering, nämlich des Zustandsmusters, beschrieben. Darüber hinaus wurden weitere mögliche Einsatzbereiche und -zwecke von Emendo diskutiert. Das folgende Kapitel fasst die wesentlichen Erkenntnisse vorliegender Arbeit zusammen und gibt einen Überblick über mögliche Weiterentwicklungen von Emendo und damit assoziierte Forschungsarbeiten.

---

## ZUSAMMENFASSUNG UND AUSBLICK

---

### 9.1 ZUSAMMENFASSUNG DER WESENTLICHEN ERGEBNISSE DIESER ARBEIT

Im Folgenden soll die vorliegende Arbeit zusammengefasst und die Quintessenz der einzelnen Kapitel herausgestellt werden.

#### 9.1.1 *Betrachtung von Gamification und Potentiale für die kompetenzorientierte Lehre*

Kapitel 2 widmet sich unterschiedlichen Sichtweisen aus den Gamification-Ansatz. Es zeigt zunächst den definitorischen Rahmen dieses Ansatzes auf und geht dabei auf die vorherrschende Begriffsheterogenität ein, die auch im Begriffsstamm *Game* vorherrscht. Die darauffolgende Einordnung von Gamification anhand der Matrix von Deterding (2016, S. 105ff.) grenzt das Konzept von Serious Games, Serious Toy Design und Toyful Design, jeweilig gestützt durch Beispiele, ab. Die zusätzliche Sichtweise Gamification durch seine wesentlichen Bestandteile, die Game Design-Elemente, vertieft das Verständnis des Ansatzes. Relevante und aktuell in der Literatur befindliche Klassifikationsschemata werden präsentiert, diskutiert und miteinander verglichen. Schließlich erweist sich das Gamification Toolkit Schema von Werbach u. Hunter (2015) als vielversprechende Grundlage für die Anwendung in einem kompetenzorientierten Lehr-Lernkontext. Letzterer wird anhand typischer Merkmale charakterisiert und die Dynamics des Gamification Toolkit als Verbindung zwischen kompetenzorientierter Lehre und dem Gamification-Ansatz theoretisch identifiziert. Diese Theorie wird gemäß dem DSR als Artefakt validiert. Hierfür werden zwei Hilfsmittel entwickelt, der GCDP, der als prozesshafte Grundlage für die Definition kompetenzorientierter und spielifizierter Lehr-Lernarrangements gilt, sowie der GCDR, welcher den GCDP systematisch instanziiert. Neben der Theorie, werden sowohl der GCDP, als auch

der GCDR in einer experimentellen Feedback-Studie validiert. In Letzterer wurde auf Basis der Theorie und mit beiden Werkzeugen ein spielifiziertes Lehr-Lernarrangement entwickelt, in der Lehre implementiert und schließlich validiert. Die Ergebnisse zeigen, dass sowohl die Theorie als auch die beiden Werkzeuge sich als tauglich für die Entwicklung derartiger Lehr-Lernarrangements gezeigt haben und bilden somit die Grundlage für weitere Forschungsarbeiten.

### **Stichpunktartige Zusammenfassung der wichtigsten Ergebnisse**

- Theoretische und Literatur-basierte Betrachtung des Gamification-Ansatzes und relevanter struktureller Bestandteile.
- Theorieentwicklung zur Vereinbarkeit von kompetenzorientierter Lehre und Gamification Design-Elemente.
- Entwicklung von GCDP und GCDR als Werkzeugunterstützung beim Design von spielifizierten und kompetenzorientierten Lehr-Lernarrangements.
- Validierung der Theorie und der Designwerkzeuge durch ein Lehrexperiment und einer experimentellen Feedback-Studie.

#### *9.1.2 Motivationstheoretische Sicht auf Gamification*

Bevor jedoch die Systematisierung bei der Entwicklung von spielifizierten und kompetenzorientierten Lehr-Lernarrangements weiter diskutiert wird, bedarf es neben einer strukturellen und definitorischen Sicht auf Gamification eine Weitere. Die motivationstheoretische Perspektive auf Gamification zeigt, auf welchen einschlägigen Theorien der Ansatz basiert. Diesen Fragestellungen und Themen widmet sich das Kapitel 3. Es zeigt die unterschiedlichen Facetten des Motivationsbegriffs, grenzt Motivationsarten voneinander ab und diskutiert für Gamification relevante einschlägige Motivationstheorien, wie die Flow-Theorie oder die SDT. Darüber hinaus wird Motivation in einem Lernkontext betrachtet und das Rahmenmodell von Rheinberg u. Fries (1998) vorgestellt, welches ein Verständnis für das Konzept der Lernmotivation erzeugt. Gleichzeitig eröffnet das Modell weitere Anknüpfungspunkte, vor allen Dingen, was die Fragestellung der gezielten Förderung von Lernmotivation angeht. Die Förderung von Lernmotivation wird zunächst anhand etablierter Prinzipien diskutiert und es werden schließlich Prinzipien aufgezeigt,

die zuvor vorgestellte Motivationstheorien berücksichtigen und die Lernmotivation in Verbindung mit Gamification bei Lernenden steigern können. Diese Prinzipien sind zudem in ein *Manifest für spielifizierte Lehre* (Bartel et al., 2017c) zusammengefasst. Darüber hinaus werden erste Forschungsdesiderate aus der Literatur und aus motivationaler Designsicht auf Gamification aufgezeigt. Diese Forschungsdesiderate werden im darauffolgenden Kapitel weiter vertieft und münden schließlich in der systematischen Erfassung des IST-Zustands der Forschung in Form von empirischen Studien.

### **Stichpunktartige Zusammenfassung der wichtigsten Ergebnisse**

- Theoretische und Literatur-basierte Betrachtung von motivationstheoretischen Grundlagen und für Gamification relevante Motivationsarten und -theorien.
- Theoretische und Literatur-basierte Betrachtung von Lernmotivation als spezielle Motivationsart.
- Entwicklung von allgemeinen Designheuristiken zur Förderung der Lernmotivation von Lernenden.
- Entwicklung von Designheuristiken zur Förderung der Lernmotivation von Lernenden durch Gamification im Speziellen.

#### *9.1.3 Empirische Studien zu Gamification in der Lehre*

Das Kapitel 4 zeigt zwei empirische Studien, welche den Forschungs- und Entwicklungsstand von Gamificationkonzepten in der Lehre fokussieren und als Domänenanalyse für die systematische Entwicklung einer domänenspezifischen Modellierungssprache dienen. Relevante Untersuchungen, bezogen auf die Dokumentenanalyse, werden ebenso beschrieben wie die zu Grunde gelegte Methodik, die aus einer Kombination von Ansätzen aus Cooper und Mayring besteht. Zudem wird die eigentliche Durchführung der Dokumentenanalyse detailliert sowie eine erste Ergebnisübersicht über die darin identifizierten Gamificationkonzepte gegeben. Ebenfalls werden Maßnahmen zur Qualitätssicherung aufgezeigt und diskutiert wie diese umgesetzt wurden. Die Ergebnisse der Dokumentenanalyse dienten als Input für die darauffolgende Lernplattformanalyse, welche die theoretische Sichtweise auf relevante Gamificationkonzepte um eine praktisch-orientierte Perspektive erweiterte. Zudem konnten dadurch die bereits identifizierten Gamificationkonzepte

verfestigt und um neue Eigenschaften ergänzt werden. Die Lernplattformanalyse wurde an die Dokumentenanalyse angelehnt durchgeführt und anhand eines strukturierten Schemas weitere einschlägige Gamificationkonzepte erfasst. Zudem wurden Lernelemente exzerpiert, die in Kombination mit Gamificationkonzepten auftraten.

Darüber hinaus stellt dieses Kapitel zwei Forschungsfragen, die im Folgenden beantwortet werden sollen.

**F.1.** Welche Elemente von Gamification werden in Lehr-Lernkontexten eingesetzt?

Beide Analysen führten zu konkreten Anforderungen, repräsentiert durch  $n=91$  Beschreibungsscharakteristika von  $n=14$  relevanten Gamification- bzw. Lernkonzepten (siehe als Übersicht Tabelle 15). Die Erhebung der Anforderungen erfolgte auf Basis einer systematischen Dokumentenanalyse, angereichert durch eine Lernplattformanalyse.

**F.2.** Wie können die identifizierten Elemente von Gamification strukturell beschrieben werden?

Diese Forschungsfrage inkl. der in Kapitel 4 aufgeführten Subforschungsfragen wird durch die Tabelle 15 beantwortet, welche alle Attribute, Zusammenhänge des Anwendungskontextes und Regeln von relevanten Domänenkonzepten systematisch aufzeigt. Ausführliche Beschreibungen und entsprechende Angaben über deren Auftreten der einzelnen Untersuchungsaspekte, befinden sich ebenfalls in Kapitel 4.

### **Stichpunktartige Zusammenfassung der wichtigsten Ergebnisse**

- Systematische und durch einschlägige Literatur gestützte Beschreibung und Durchführung einer Dokumenten- und Lernplattformanalyse zur Analyse der Domäne.
- Erfassung der strukturellen Eigenschaften, Zusammenhänge und Regeln von Gamification- und Lernkonzepten.

#### **9.1.4 *Emendo DSML***

Das Kapitel 5 gibt einen Überblick über notwendige Grundlagen von domänenspezifischen Sprachen als Teil des Domain-specific Modeling-Ansatzes (Kelly u. Tolvanen, 2008). Die Bestandteile von Letzterem werden ebenso detailliert und auf Basis dessen Vorüberlegungen für die zu erstellende Emendo DSML diskutiert. Diese beinhaltend das Vorgehen, die Zielsetzung bzw. der Anwendungsbereich der DSML sowie generische Anforderungen. Darauffol-



gend werden die aus Kapitel 4 erfassten Domänenkonzepte abstrahiert und verdichtet und so eine abstrakte Syntax der DSML sukzessive entwickelt und beschrieben. Zudem erfolgte die Entwicklung der konkreten Syntax auf Basis einer Nutzerstudie, die zum Ziel hatte, möglichst hohe Übereinstimmungen von Domänenkonzept zu grafischer Repräsentation zwischen potentiellen Nutzern zu identifizieren, um daraus eine konkrete Syntax abzuleiten.

Durch die Ausführungen in Kapitel 5 wird zudem die eingangs gestellte Forschungsfrage **F.3.** (*Wie sieht eine aus den empirischen Daten abgeleitete Systematisierung in Form einer grafischen DSML aus?*) beantwortet.

### **Stichpunktartige Zusammenfassung der wichtigsten Ergebnisse**

- Literatur-gestützter Überblick zu domänenspezifischen Sprachen.
- Detaillierte Beschreibung des DSM-Ansatzes und seiner Bestandteile.
- Definition des Vorgehens, der Zielsetzung sowie generischer Anforderungen an die Emendo DSML.
- Konzeptuelle Entwicklung der abstrakten Syntax.
- Empirisch-gestützte Entwicklung der konkreten Syntax.

#### *9.1.5 Emendo Designer und Generator*

Das darauffolgende Kapitel 6 beschreibt die Frameworks EMF und Sirius, welche für die Umsetzung der im vorherigen Kapitel konzeptuell entwickelten Emendo DSML verwendet wurden. Es zeigt zudem relevante Modellierungsaspekte und -herausforderungen, die sich bei der Implementierung des Emendo Designers zeigten und wie deren Umsetzung erfolgte. Zudem befasst sich das Kapitel mit der Erstellung des Emendo Generators, welcher als Teil des Emendo Designers, Modellinstanzen der Emendo DSML in Quellcode transformiert und in das Emendo LMS integriert. Abschließend werden zu der Umsetzung verwandte Arbeiten präsentiert und diskutiert und aufgezeigt, dass zum Zeitpunkt der Anfertigung dieser Arbeit kein vergleichbares System existiert.

### **Stichpunktartige Zusammenfassung der wichtigsten Ergebnisse**

- Literatur-basierter Überblick über EMF und Sirius als für die Umsetzung der Emendo DSML und des Emendo Designers relevante Frameworks.

- Beschreibung der Implementierung des Emendo Designers.
- Beschreibung des Emendo Generators als Bestandteil des Emendo Designers.
- Diskussion zu Emendo verwandter Arbeiten.

#### 9.1.6 *Emendo Learning Management System*

Das Kapitel 7 fokussiert das Emendo LMS und stellt dieses aus unterschiedlichen Sichten vor. Es folgt zunächst eine Anforderungsanalyse, die funktionale und nichtfunktionale Anforderungen an das LMS definiert und in Use Cases überführt. Darauf aufbauend wurde die Systemarchitektur entworfen, wobei MVC als Architekturmuster angewandt wurde. Für das LMS zentrale Komponenten werden anhand unterschiedlicher UML-Diagramme detailliert beschrieben und deren Funktionsweise erläutert. Hinzukommend werden nicht in der DSML umgesetzte und aus Kapitel 4 stammende Domänenkonzepte beschrieben. Ebenfalls wird auf die Entwicklung der GUI eingegangen und relevante Bausteine dieser demonstriert. Schließlich wird das Mobile Emendo LMS als eine weitere Plattform vorgestellt und damit demonstriert, dass das LMS die nichtfunktionale Anforderung der Portabilität erfüllt.

Bezogen auf die eingangs gestellte 4. Forschungsfrage (*Wie sieht eine tool-gestützte Umsetzung der erstellten Systematik im Hochschulkontext am Beispiel von ausgewählten Themen des Software Engineerings aus?*) kann festgestellt werden, dass diese Frage durch die Kapitel 6 und 7 beantwortet wird. Beide Kapitel schließen die Beschreibungen der Emendo Werkzeugkette ein und detaillieren die einzelnen Bestandteile.

#### **Stichpunktartige Zusammenfassung der wichtigsten Ergebnisse**

- Systematische Anforderungserhebung für die Entwicklung des Emendo LMS.
- Beschreibung der Systemarchitektur samt relevanter Komponenten unter Verwendung der UML.
- Beschreibung der GUI des LMS und der Benutzerinteraktionsmöglichkeiten.
- Vorstellung des Mobile Emendo LMS als Proof-of-Concept für die Portabilität des Emendo LMS.

### 9.1.7 *Evaluation*

Motiviert durch den DSR-Ansatz, der als Forschungsdesign dieser Arbeit zu Grunde liegt, zeigt Kapitel 8 zwei Evaluationen, welche die entwickelten Artefakte beleuchten. In einer qualitativ angelegten Feldstudie mit Lehrenden und damit potentiell zukünftigen Nutzern von Emendo, konnte gezeigt werden, dass Emendo generell positiv aufgenommen wird. Auch das durch Probanden umgesetzte Modellierungsszenario wurde positiv beurteilt, auch wenn sich Verbesserungsmöglichkeiten durch die Aussagen der Probanden identifizieren lassen konnten, die vor allem die DSML und den Designer betreffen. In einer weiteren Evaluation wurden die generischen Anforderungen der DSML aus Kapitel 5 auf deren Erfüllungsgrad hin überprüft. Es konnte gezeigt werden, dass alle generischen Anforderungen umgesetzt werden konnten. Schließlich wurde abschließend ein Einsatzszenario von Emendo aus Sicht von Dozierenden vorgestellt und beschrieben, welches die Methode spielifizierte JiTT einsetzt.

### **Stichpunktartige Zusammenfassung der wichtigsten Ergebnisse**

- Beschreibung einer qualitativen Feldstudie zur Evaluation des Emendo Designers und der darin eingebetteten Emendo DSML.
- Abgleich der generischen Anforderungen an die DSML mit der tatsächlichen Umsetzung.
- Beispielhafte Demonstration eines Einsatzes von Emendo für die Konzeption einer spielifizierten JiTT-Lerneinheit.

## 9.2 AUSBLICK UND WEITERE FORSCHUNGSARBEITEN

Der vorherige Abschnitt fasste die vorliegende Arbeit zusammen und exzerpierte relevante Beiträge, welche diese Arbeit leistet. Darüber hinaus existieren Anknüpfungspunkte für weitere Forschungsarbeiten oder Studien, die im Folgenden kurz vorgestellt werden. Einige der folgenden Anknüpfungspunkte ergeben sich unter anderem durch die Aussagen von Personen, die an der Evaluation von Emendo, beschrieben in Kapitel 8, teilnahmen.

### 9.2.1 *Portierung als Webanwendung*

Sowohl der Emendo Designer als auch das Emendo LMS könnten von dem jetzigen Stand einer lokal ausführbaren Client-Anwendungskette in einen Webbrowser portiert und damit zu einer 3-Tier-Architektur weiterentwickelt werden. Dieser Schritt würde den Einsatz auf unterschiedlichen Betriebssystemlandschaften ohne Java Virtual Machine (JVM) erlauben, welche über einen Webbrowser mitsamt notwendigen Webtechnologien, wie beispielsweise die Ausführung von Javascript, verfügen und diese Ausführung erlauben. Als Folge dieser Portierungsarbeit wäre der Zugang zu Emendo für einen größeren Interessentenkreis sowohl auf Lehrenden- als auch Studierendenseite möglich. Ebenso könnten parallel mehrere Lerneinheiten im Emendo LMS angeboten werden, die von Lernenden auch mobil als App bearbeitet werden können. Neben der bereits erstellten mobilen Version des Emendo LMS auf Basis von Android wäre zu überlegen weitere mobile Plattformen, wie iOS, zu unterstützen.

### 9.2.2 *Weiterentwicklung der Notation des Emendo Designer bzw. der Emendo DSML*

Einhergehend mit der Portierung in eine Webumgebung des Emendo Designers kann die bestehende Notation erweitert werden, um noch stärker das tatsächlich dargestellte Endprodukt einer Lerneinheit im Emendo LMS darzustellen. In der durchgeführten Evaluation wurde geäußert, dass es für Dozierende unter Umständen schwer vorstellbar sei, wie das Modell sich tatsächlich im Emendo LMS darstellen lässt. Durch die Modularität und die damit verbundene Komponentenorientierung, welche beide im Emendo LMS zum Tragen kommen, wäre es möglich, die Notation der Emendo DSML so anzupassen, dass die einzelnen Inhaltsseiten des Emendo LMS direkt als Notationselemente der DSML im Designer erzeugt werden. Das Prinzip dahinter wäre vergleichbar mit einem Sketching einer GUI, welche in diesem Fall nicht nur für die Erstellung eines Prototypen herangezogen wird und händisch codiert oder mit Window Buildern modelliert werden muss, sondern vielmehr die einzelnen Module des LMS als Widgets vorliegen, die auf Knopfdruck das gewünschte Artefakt nach Maßstäben einschlägiger Usability Anwendungs-bereit erzeugen und für Dozierende im Designer direkt editierbar machen. Der Emendo Designer würde somit zu einem What You See Is What You Get (WYSIWYG) Editor weiterentwickelt.

### 9.2.3 Anbindung an bestehende Learning Management Systeme

Das frei verfügbare Kursmanagementsystem Moodle<sup>1</sup> ist eines der am weltweit verbreitetsten mit annähernd 93.000 dokumentierten Installation<sup>2</sup> in über 230 Ländern. Zwar existieren für Moodle Plug-Ins, welche Aspekte von Gamification in die Lernplattform integrieren lassen, wie z. B. *Level up!*<sup>3</sup>, *Interactive Content*<sup>4</sup>, *Game*<sup>5</sup> oder *Quizventure*<sup>6</sup> jedoch ist deren Funktionalität sowie wissenschaftliche Fundierung in Bezug auf Gamification sehr beschränkt. Beispielsweise sieht das Plug-In *Game* die bloße Existenz von Quizzes als eine Form der Spielifizierung, was nicht der hier begründeten wissenschaftlichen Auffassung von Spielifizierung entspricht. Auch konzentrieren sich vorhandene Plug-Ins auf das Sammeln einzelner Achievements, wie beispielsweise *Level up!* (vornehmlich Level und Punkte), was die Möglichkeiten für den effektiven Einsatz von Gamification stark einschränkt. Eine Kombination von unterschiedlichen Plug-Ins ist zwar beispielsweise für Moodle dokumentiert (Roderus, 2015), geht jedoch mit einem hohen Aufwand für die Erstellung derartiger Lerneinheiten einher. Daher stellt sich die Frage einer Vereinbarkeit bzw. Integration von Emendo in vorhandene, weit verbreitete LMS, wie Moodle. Grundsätzlich wären Möglichkeiten zur Integration der Emendo Plattform über Sharable Content Object Reference Model (SCORM), Learning Object Metadata (LOM) oder Question & Test Interoperability (QTI) denkbar. Beispielsweise könnte für die unterschiedlichen Zielplattformen und Zielformate jeweils ein Generator verfasst werden, der das Mapping der Konzepte aus der Emendo DSML und den daraus entstandenen Modellen auf die Austauschformate übernimmt. Obwohl SCORM als einziges Austauschformat explizit die Beschreibung von Lernpfaden in Form von Aktivitätsbäumen vorsieht, so ist es jedoch nicht ohne weiteres möglich das Regelkonzept in die angedeuteten Austauschformate zu integrieren, da es komplexere und umfangreichere Formulierungen erlaubt, die aus Architektursicht tiefere Veränderungen in einem LMS, wie Moodle, zur Folge hätten und aktuell darin nicht über die Plug-In Schnittstellen umgesetzt werden können. Des weiteren könnte eine Integration auf Datenebene erfolgen, in dem die Metadaten der Lernbausteine und deren Bearbeitungszustände aus der Datenbank extrahiert und in ein LMS, wie Moodle, importiert werden. Nichtsdestotrotz stellt die Integration in

1 Siehe <https://moodle.net/>, abgerufen am: 01.02.2018.

2 Die Daten wurden erhoben am 01.02.2018 und stammen von der Moodle Statistik-Seite <https://moodle.net/sites/>.

3 Siehe [https://moodle.org/plugins/block\\_xp](https://moodle.org/plugins/block_xp), abgerufen am: 01.02.2018.

4 Siehe [https://moodle.org/plugins/mod\\_hvp](https://moodle.org/plugins/mod_hvp), abgerufen am: 01.02.2018.

5 Siehe [https://moodle.org/plugins/mod\\_game](https://moodle.org/plugins/mod_game), abgerufen am: 01.02.2018.

6 Siehe [https://moodle.org/plugins/mod\\_quizgame](https://moodle.org/plugins/mod_quizgame), abgerufen am: 01.02.2018.

bestehende LMS, wie Moodle, eine weitere Forschungsarbeit dar, die es aufgrund der Häufigkeiten an Äußerungen von an der Evaluation aus Kapitel 8 teilnehmenden Personen zu berücksichtigen gilt.

#### 9.2.4 *Adaptivität des Emendo LMS*

Der Gedanke, das Emendo LMS dahingehend anzupassen, dass die Schwierigkeit noch zu bearbeitenden Lernbausteine intelligent und adaptiv vom LMS auf Basis bereits erfasster Learning Analytics zu verändern, wurde bereits im Zusammenhang mit dem Architekturmuster MVC geäußert. Dieser Gedanke könnte noch weiter gefasst werden und das Emendo LMS nicht nur adaptiv in Bezug auf Lernbausteine und damit auf Inhalte sein, sondern auch auf die eingesetzten Game Design-Elemente. Es könnte neben Learning Analytics ebenso Metriken erfasst werden, die Aussagen darüber erlauben, inwiefern einzelne Game Design-Elemente von Lernenden angenommen werden. Hier steckt die Forschung noch in den Anfängen, wie Arbeiten von Schöbel et al. (2016) bzw. Schöbel u. Söllner (2016) zeigen. Um diese Beziehung zwischen Lernenden und Game Design-Elemente individueller zu gestalten, muss zunächst dieses Verhältnis genauer erforscht werden. Auch wenn hierzu bereits Arbeiten existieren (Sailer, 2016; Mazarakis, 2017; Mekler et al., 2017) gilt es diesen Forschungsbereich durch mehr Empirie weiter auszubauen. Emendo könnte hierfür ein Werkzeug sein, dass derartige Forschung weiter unterstützt.

---

## LITERATURVERZEICHNIS

---

- [Aarseth 2001] AARSETH, Espen: Computer Game Studies, Year One. In: *The International Journal of Computer Game Research* 1 (2001), Nr. 1. <http://www.gamestudies.org/0101/editorial.html>, Abruf: 18.04.2018
- [Abke et al. 2012a] ABKE, Jörg; BRUNE, Phillip; HAUPT, Wolfram; LANDES, Dieter; PFEIFFER, Volkhard; SEDELMAIER, Yvonne; HAGEL, Georg; MOTTOK, Jürgen; NIEMETZ, Michael; SCHROLL-DECKER, Irmgard; STUDT, Reimer: EVELIN – ein Forschungsprojekt zur systematischen Verbesserung des Lernens von Software Engineering. In: *Embedded Software Engineering Kongress 2012*, Elektronikpraxis/Vogel, 2012, S. 653–658
- [Abke et al. 2012b] ABKE, Jörg; BRUNE, Phillip; HAUPT, Wolfram; LANDES, Dieter; PFEIFFER, Volkhard.; SEDELMAIER, Yvonne; HAGEL, Georg; MOTTOK, Jürgen; NIEMETZ, Michael; SCHROLL-DECKER, Irmgard; STUDT, Reimer: *Projektantrag EVELIN - Experimentelle Verbesserung des Lernens von Software Engineering*. Coburg, 2012
- [Abraham u. Müller 2009] ABRAHAM, Ulf; MÜLLER, Astrid: Aus Leistungsaufgaben lernen. In: *Praxis Deutsch* 36 (2009), Nr. 214, S. 4–12
- [Adams u. Rollings 2007] ADAMS, Ernest; ROLLINGS, Andrew: *Fundamentals of Game Design*. Upper Saddle River, N.J.: Pearson Prentice Hall, 2007 (Game design and development series)
- [Ahn et al. 2014] AHN, June; PELLICONE, Anthony; BUTLER, Brian S.: Open badges for education: What are the implications at the intersection of open systems and badging? In: *Research in Learning Technology* 22 (2014), Nr. 1. <http://www.researchinlearningtechnology.net/index.php/rlt/article/view/23563>, Abruf: 18.05.2015
- [Ajzen 1991] AJZEN, Icek: The theory of planned behavior. In: *Organizational Behavior and Human Decision Processes* 50 (1991), Nr. 2, S. 179–211
- [Akremi 2014] AKREMI, Leila: Stichprobenziehung in der qualitativen Sozialforschung. In: BLASIUS, Jörg (Hrsg.); BAUR, Nina (Hrsg.): *Handbuch Methoden der empirischen Sozialforschung*. Wiesbaden: Springer VS, 2014, S. 265–282

- [Al-Mutawa et al. 2014] AL-MUTAWA, Hussain A.; DIETRICH, Jens; MARSLAND, Stephen; MCCARTIN, Catherine: On the Shape of Circular Dependencies in Java Programs. In: *23rd Australian Software Engineering Conference*, IEEE, 2014, S. 48–57
- [Alves et al. 2014] ALVES, Fábio P.; MACIEL, Cristiano; ANACLETO, Júnia C.: Guidelines for the Gamification in Mobile Social Networks. In: MEISELWITZ, Gabriele (Hrsg.): *Social Computing and Social Media* Bd. 8531. Cham, Switzerland: Springer International Publishing, 2014, S. 559–570
- [Anderson et al. 2013] ANDERSON, Ashton; HUTTENLOCHER, Daniel; KLEINBERG, Jon M.; LESKOVEC, Jure: Steering user behavior with badges. In: SCHWABE, Daniel (Hrsg.); ALMEIDA, Virgilio (Hrsg.); GLASER, Hartmut (Hrsg.): *Proceedings of the 22nd international conference on World Wide Web*, International World Wide Web Conferences Steering Committee, 2013, S. 95–106
- [Anderson u. Krathwohl 2001] ANDERSON, Lorin W.; KRATHWOHL, David R.: *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives*. New York: Longman, 2001
- [Apperl u. Brenn 1991] APPERL, Josef; BRENN, Hubert: Wie Schüler den Unterricht von Lehrern und Lehrer-Studenten erleben und einschätzen. In: *Unser Weg* 46 (1991), Nr. 1, S. 6–11
- [Arbeitskreis Deutscher Qualifikationsrahmen 2003] ARBEITSKREIS DEUTSCHER QUALIFIKATIONSRAHMEN; BUNDESMINISTERIUM FÜR BILDUNG UND FORSCHUNG (Hrsg.): *Deutscher Qualifikationsrahmen für lebenslanges Lernen*. Version: 22.03.2011. [http://www.dqr.de/media/content/Deutscher\\_EQR\\_Referenzierungsbericht.pdf](http://www.dqr.de/media/content/Deutscher_EQR_Referenzierungsbericht.pdf), Abruf: 03.03.2016
- [Ašeriškis u. Damaševičius 2014] AŠERIŠKIS, Darius; DAMAŠEVIČIUS, Robertas: Gamification Patterns for Gamification Applications. In: *Procedia Computer Science* 39 (2014), 83–90. <http://www.sciencedirect.com/science/article/pii/S1877050914014318>, Abruf: 26.02.2015
- [Astrachan 2004] ASTRACHAN, Owen L.: Non-competitive programming contest problems as the basis for just-in-time teaching. In: *34th Annual Frontiers in education conference proceedings*, IEEE, 2004, S. T3H/20–T3H/24
- [Atkinson 1999] ATKINSON, Colin: Supporting and Applying the UML Conceptual Framework. In: GOOS, Gerhard (Hrsg.); HARTMANIS, Juris (Hrsg.); VAN LEEUWEN, Jan (Hrsg.); BÉZIVIN, Jean (Hrsg.); MULLER, Pierre-Alain



- (Hrsg.): *The Unified Modeling Language. «UML»'98: Beyond the Notation* Bd. 1618. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, S. 21–36
- [Bahji et al. 2013] BAHJI, Salah E.; LEFDAOUI, Youssef; EL ALAMI, Jamila: Enhancing motivation and engagement: A top-down approach for the design of a learning experience according to the S2P-LM. In: *International Journal of Emerging Technologies in Learning* 8 (2013), Nr. 6, 35–41. <http://online-journals.org/i-jet/article/view/2955>, Abruf: 19.05.2015
- [Bailey u. Forbes 2005] BAILEY, Tammy; FORBES, Jeffrey: Just-in-time Teaching for CSo. In: *Proceedings of the 36th Special Interest Group on Computer Science Education (SIGCSE) Technical Symposium on Computer Science Education*, ACM, 2005 (SIGCSE '05), S. 366–370
- [Balzert 2009] BALZERT, Helmut: *Lehrbücher der Informatik*. Bd. 1: *Basiskonzepte und Requirements Engineering*. 3. Aufl. Heidelberg: Spektrum, Akad. Verl., 2009
- [Bandler u. Grinder 1994] BANDLER, Richard; GRINDER, John: *Metasprache und Psychotherapie: Die Struktur der Magie Band I*. Bd. 11. 8. Paderborn: Junfermannsche Verlagsbuchhandlung, 1994
- [Bandura 1977] BANDURA, Albert: Self-efficacy: Toward a unifying theory of behavioral change. In: *Psychological Review* 84 (1977), Nr. 2, S. 191–215
- [Bandura 1997] BANDURA, Albert: *Self-efficacy: The exercise of control*. New York: W.H. Freeman and Company, 1997
- [Banfield u. Wilkerson 2014] BANFIELD, James; WILKERSON, Brad: Increasing Student Intrinsic Motivation And Self-Efficacy Through Gamification Pedagogy. In: *Contemporary Issues in Education Research* 7 (2014), Nr. 4, 291–298. <http://cluteinstitute.com/ojs/index.php/CIER/article/view/8843>, Abruf: 04.03.2015
- [Bang et al. 2009] BANG, Magnus; SVAHN, Mattias; GUSTAFSSON, Anton: Persuasive design of a mobile energy conservation game with direct feedback and social cues. In: *Proceedings of Digital Games Research Association (DiGRA)*, 2009, S. 1–7
- [Barata et al. 2013] BARATA, Gabriel; GAMA, Sandra; JORGE, Joaquim; GONÇALVES, Daniel: Engaging engineering students with gamification. In: GATZIDIS, Christos (Hrsg.); ANDERSON, Eike (Hrsg.): *2013 5th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)*, IEEE, 2013, S. 1–8

- [Bartel et al. 2014a] BARTEL, Alexander; FIGAS, Paula; HAGEL, Georg: Mobile Game-Based Learning in University Education. In: FELLER, Sebastian (Hrsg.); YENGİN, Ilker (Hrsg.): *Educating in Dialog*. Amsterdam/Philadelphia: John Benjamins Publishing Company, 2014, S. 159–180
- [Bartel et al. 2014b] BARTEL, Alexander; FIGAS, Paula; HAGEL, Georg: Using a Scenario-Based Approach for Learning Software Engineering. In: HAGEL, Georg (Hrsg.); MOTOK, Jürgen (Hrsg.): *Proceedings of ECSEE 2014 - European Conference Software Engineering Education*, Shaker, 2014, S. 167–179
- [Bartel et al. 2015] BARTEL, Alexander; FIGAS, Paula; HAGEL, Georg: Towards a Competency-based Education with Gamification Design Elements. In: *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play*, ACM, 2015, S. 457–462
- [Bartel et al. 2016] BARTEL, Alexander; FIGAS, Paula; WECKERLE, Birgit; WINKLER, Katrin; HAGEL, Georg: Spielifizierte Lernaufgaben im Blended Learning Kontext. In: *Medienproduktion - Online-Zeitschrift für Wissenschaft und Praxis* (2016), Nr. 10, 6–9. [http://www2.tu-ilmenau.de/zsmp/wordpress/wp-content/A10/2\\_Bartel.pdf](http://www2.tu-ilmenau.de/zsmp/wordpress/wp-content/A10/2_Bartel.pdf), Abruf: 31.12.2016
- [Bartel u. Hagel 2014] BARTEL, Alexander; HAGEL, Georg: Engaging Students With a Mobile Game-Based Learning System in University Education. In: *International Journal of Interactive Mobile Technologies* 8 (2014), Nr. 4, 56–60. <http://online-journals.org/index.php/i-jim/article/download/3991/3298>, Abruf: 18.04.2018
- [Bartel u. Hagel 2016a] BARTEL, Alexander; HAGEL, Georg: Gamified Just-in-Time Teaching - A Conceptual Approach Based on Best Practices. In: HAGEL, Georg (Hrsg.); MOTOK, Jürgen (Hrsg.): *Proceedings of ECSEE 2016 - European Conference Software Engineering Education*, Shaker, 2016, S. 1–17
- [Bartel u. Hagel 2016b] BARTEL, Alexander; HAGEL, Georg: Gamifying the Learning of Design Patterns in Software Engineering Education. In: *Proceedings of Global Engineering Education Conference (EDUCON)*, IEEE, 2016, S. 74–79
- [Bartel et al. 2017a] BARTEL, Alexander; HAGEL, Georg; WOLFF, Christian: Effective Integration of Gamification and Learning Management Systems for Creating Gamified Learning Arrangements. In: DOMENECH, Josep (Hrsg.); VICENT-VELA, Cinta M. (Hrsg.); DE LA POZA, Elena (Hrsg.); BLAZQUEZ, Desamparados (Hrsg.): *Proceedings of the 3rd International Conference on Higher Education Advances*, 2017, S. 679–686

- [Bartel et al. 2017b] BARTEL, Alexander; HAGEL, Georg; WOLFF, Christian: Towards a Generic Platform for Implementing Gamified Learning Arrangements in Engineering Education. In: *Proceedings of IEEE Global Engineering Education Conference (EDUCON)*, IEEE, 2017, S. 1501–1504
- [Bartel et al. 2018] BARTEL, Alexander; HAGEL, Georg; WOLFF, Christian: Emendo - A Toolchain for Creating Gamified Learning Arrangements for Online Learning Settings (zur Veröffentlichung angenommen). In: *Proceedings of the 4rd International Conference on Higher Education Advances*, 2018
- [Bartel et al. 2017c] BARTEL, Alexander; SOSKA, Alexander; WOLFF, Christian; HAGEL, Georg; MOTTOK, Jürgen: Entwicklung eines Manifests für spielifizierte Hochschullehre. In: BURGHARDT, Manuel (Hrsg.); WIMMER, Raphael (Hrsg.); WOLFF, Christian (Hrsg.); WOMSER-HACKER, Christa (Hrsg.): *Mensch und Computer 2017 - Tagungsband*, Gesellschaft für Informatik e.V, 2017, S. 299–304
- [Bartel et al. 2017d] BARTEL, Alexander; SOSKA, Alexander; WOLFF, Christian; HAGEL, Georg; MOTTOK, Jürgen; BMBF-PROJEKT EVELIN (Hrsg.): *Manifest für spielifizierte Hochschullehre*. <http://gamefulmanifesto.org/>. Version: 2017, Abruf: 14.09.2017
- [Bartle 1996] BARTLE, Richard: Hearts, clubs, diamonds, spades: Players who suit MUDs. In: *Journal of MUD research* 1 (1996), Nr. 1
- [Bedwell et al. 2012] BEDWELL, Wendy L.; PAVLAS, Davin; HEYNE, Kyle; LAZZARA, Elizabeth H.; SALAS, Eduardo: Toward a taxonomy linking game attributes to learning an empirical study. In: *Simulation & Gaming* 43 (2012), Nr. 6, 729–760. <http://sag.sagepub.com/content/43/6/729.short>, Abruf: 18.04.2018
- [Berkling u. Thomas 2013] BERKLING, Kay; THOMAS, Christoph: Gamification of a Software Engineering course and a detailed analysis of the factors that lead to it's failure. In: *International Conference on Interactive Collaborative Learning (ICL)*, IEEE, 2013, S. 525–530
- [Berlyne 1960] BERLYNE, D. E.: *Conflict, Arousal and Curiosity*. New York: McGraw-Hill Publishing Company Ltd., 1960
- [Bertoa u. Vallecillo 2010] BERTO A, Manuel F.; VALLECILLO, Antonio: Quality Attributes for Metamodels. In: *3th TOOLS Workshop on Quantitative Approaches in Object-Oriented Software Engineering*, 2010

- [Betts et al. 2013] BETTS, Ben W.; BAL, Jay; BETTS, Alan W.: Gamification as a tool for increasing the depth of student understanding using a collaborative e-learning environment. In: *International Journal of Continuing Engineering Education and Life Long Learning* 23 (2013), Nr. 3, S. 213–228
- [Bjork u. Holopainen 2005] BJORK, Staffan; HOLOPAINEN, Jussi: *Patterns in game design*. 1st ed. Hingham, Mass.: Charles River Media, 2005
- [Bloemen u. Schlömer 2012] BLOEMEN, André; SCHLÖMER, Tobias: Berufliche Handlungskompetenz. In: PAECHTER, Manuela (Hrsg.); STOCK, Michaela (Hrsg.); SCHMÖLZER-EIBINGER, Sabine (Hrsg.); SLEPCEVIC-ZACH, Peter (Hrsg.); WEIRER, Wolfgang (Hrsg.): *Handbuch Kompetenzorientierter Unterricht*. Weinheim, Bergstr: Beltz, 2012 (Pädagogik), S. 119–139
- [Bloom 1984] BLOOM, Benjamin S.: The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective One-to-One Tutoring. In: *Educational Researcher* 13 (1984), Nr. 6, 4–16. <http://web.mit.edu/5.95/readings/bloom-two-sigma.pdf>, Abruf: 18.04.2018
- [Bloom et al. 1956] BLOOM, Benjamin S.; ENGELHART, Max D.; FURST, Edward J.; HILL, Walker H.; KRATHWOHL, David R.: *Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain*. New York, USA: David McKay, 1956
- [Bogost 2011] BOGOST, Ian: Persuasive games: exploitationware. In: *Game-sutra* 9 (2011), 2013. [http://www.gamasutra.com/view/feature/134735/persuasive\\_games\\_exploitationware.php?print=1](http://www.gamasutra.com/view/feature/134735/persuasive_games_exploitationware.php?print=1), Abruf: 18.04.2018
- [Bönsch et al. 2010] BÖNSCH, Manfred; KOHNEN, Helga; MÖLLERS, Birgit; MÜLLER, Günter; NATHER, Wolfgang; SCHÜÜRMAN, Anja: *Kompetenzorientierter Unterricht*. Braunschweig: Westermann, 2010
- [Boote u. Beile 2005] BOOTE, David.; BEILE, Penny: Scholars before researchers: On the centrality of the dissertation literature review in research preparation. In: *Educational Researcher* 34 (2005), Nr. 6, S. 3–15
- [Böttcher et al. 2015] BÖTTCHER, Axel; KÄMPER, Andreas; THURNER, Veronika: On Analyzing the Effectiveness of Just-in-Time Teaching. In: *Proceedings of IEEE Global Engineering Education Conference (EDUCON)*, IEEE, 2015, S. 453–461
- [Boulet 2012] BOULET, Guy: Gamification: The Latest Buzzword and the Next Fad. In: *Elearn Magazine* (2012), Nr. 12. <https://elearnmag.acm.org/archive.cfm?aid=2421596>, Abruf: 18.02.2015

- [Brambilla et al. 2012] BRAMBILLA, Marco; CABOT, Jordi; WIMMER, Manuel: *Synthesis Lectures on Software Engineering*. Bd. 1: *Model-Driven Software Engineering in Practice*. 1. San Rafael: Morgan & Claypool Publishers, 2012
- [Brambilla et al. 2017] BRAMBILLA, Marco; CABOT, Jordi; WIMMER, Manuel: *Synthesis Lectures on Software Engineering*. Bd. 4: *Model-driven software engineering in practice*. 2. San Rafael: Morgan & Claypool Publishers, 2017
- [Brereton et al. 2007] BRERETON, Pearl; KITCHENHAM, Barbara A.; BUDGEN, David; TURNER, Mark; KHALIL, Mohamed: Lessons from applying the systematic literature review process within the software engineering domain. In: *Journal of Systems and Software* 80 (2007), Nr. 4, S. 571–583
- [Bringhurst 2008] BRINGHURST, Robert: *The Elements of Typographic Style: Version 3.2*. Point Roberts, WA, USA: Hartley & Marks Publishers, 2008
- [Brophy 2010] BROPHY, Jere: *Motivating students to learn*. 3. New York: Routledge, 2010
- [Burger 2014] BURGER, Erik: *Flexible Views for View-based Model-driven Development*. Karlsruhe, Karlsruher Institut für Technologie, Diss., 2014
- [Burke 2014a] BURKE, Brian: *Gamify: How gamification motivates people to do extraordinary things*. Brookline, MA: Bibliomotion, 2014
- [Burke 2014b] BURKE, Brian: *Gartner Redefines Gamification*. [http://blogs.gartner.com/brian\\_burke/2014/04/04/gartner-redefines-gamification/](http://blogs.gartner.com/brian_burke/2014/04/04/gartner-redefines-gamification/). Version: 2014, Abruf: 05.06.2015
- [Buschmann et al. 2007] BUSCHMANN, Frank; HENNEY, Kevlin; SCHMIDT, Douglas C.: *Wiley series in software design patterns*. Bd. 4: *Pattern-oriented software architecture: A pattern language for distributed computing*. Chichester: Wiley, 2007
- [Buschmann et al. 1996] BUSCHMANN, Frank; MEUNIER, Regine; ROHNERT, Hans; SOMMERLAD, Peter; STAL, Michael: *Pattern-oriented software architecture: A system of patterns*. Bd. 1. Chichester: Wiley, 1996
- [Byrne 2005] BYRNE, Edward: *Game Level Design*. 1. Hingham, Mass.: Charles River Media, 2005 (Charles River Media game development series)
- [Caillois 2001] CAILLOIS, Roger: *Man, play, and games*. Urbana: University of Illinois Press, 2001

- [Calvillo-Gómez et al. 2010] CALVILLO-GÁMEZ, Eduardo H.; CAIRNS, Paul; COX, Anna L.: Assessing the Core Elements of the Gaming Experience. In: BERNHAUPT, Regina (Hrsg.): *Evaluating user experience in games*. London, UK: Springer London, 2010, S. 47–71
- [Camp et al. 2010] CAMP, Mary E.; MIDDENDORF, Joan; SULLIVAN, Carol S.: Using Just-in-Time Teaching to Motivate Student Learning. In: SIMKINS, Scott (Hrsg.); MAIER, Mark H. (Hrsg.): *Just-in-Time Teaching*. Sterling, Virginia: Stylus Publishing, 2010, S. 25–38
- [Capernaum Ltd. 2016] CAPERNAUM LTD.: *ALISON*. <https://alison.com/>. Version: 2016, Abruf: 25.06.2016
- [Caponetto et al. 2014] CAPONETTO, Ilaria; EARP, Jeffrey; OTT, Michela: Gamification and Education: A Literature Review. In: BUSCH, Carsten (Hrsg.): *Proceedings of the 8th European Conference on Game Based Learning*, Academic Conferences and Publishing International Limited, 2014, S. 50–57
- [Carter 2009] CARTER, Paul: An Experiment with Online Instruction and Active Learning in an Introductory Computing Course for Engineers: JiTT Meets CS1. In: *Proceedings of the 14th Western Canadian Conference on Computing Education*, ACM, 2009 (WCCCE '09), S. 103–108
- [Carter 2012] CARTER, Paul: An Experience Report: On the Use of Multimedia Pre-instruction and Just-in-time Teaching in a CS1 Course. In: *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education (SIG-CSE '12)*, ACM, 2012, S. 361–366
- [Cashman u. Eschenbach 2003] CASHMAN, Eileen M.; ESCHENBACH, Elizabeth A.: Using On-Line Quizzes Outside the Classroom to Increase Student Engagement Inside the Classroom. In: *Proceedings of Developing Tomorrow's Workforce*, 2003
- [CDU Deutschland et al. 2013] CDU DEUTSCHLAND; CSU-LANDESLEITUNG; SPD: *Deutschlands Zukunft gestalten: Koalitionsvertrag zwischen CDU, CSU und SPD - 18. Legislaturperiode*. <https://www.cdu.de/sites/default/files/media/dokumente/koalitionsvertrag.pdf>. Version: 2013, Abruf: 22.04.2016
- [Cerasoli et al. 2014] CERASOLI, Christopher P.; NICKLIN, Jessica M.; FORD, Michael T.: Intrinsic motivation and extrinsic incentives jointly predict performance: a 40-year meta-analysis. In: *Psychological bulletin* 140 (2014), Nr. 4, S. 980–1008

- [Challco et al. 2014] CHALLCO, Geiser C.; MOREIRA, Dilvan; MIZOGUCHI, Richiro; ISOTANI, Seiji: Towards an Ontology for Gamifying Collaborative Learning Scenarios. In: TRAUSAN-MATU, Stefan (Hrsg.); BOYER, Kristy E. (Hrsg.); CROSBY, Martha (Hrsg.); PANOURGIA, Kitty (Hrsg.): *Intelligent Tutoring Systems: 12th International Conference*, Springer, 2014 (Lecture Notes in Computer Science / Programming and Software Engineering), S. 404–409
- [Chomsky 1959] CHOMSKY, Noam: On Certain Formal Properties of Grammars. In: *Information and Control* 2 (1959), Nr. 2, S. 137–167
- [Chou 2015] CHOU, Yu-Kai: *Actionable gamification: Beyond points, badges, and leaderboards*. Colorado Springs, US: CreateSpace Independent Publishing Platform, 2015
- [Christmann et al. 1999] CHRISTMANN, Ursula; GROEBEN, Norbert; SCHREIER, Margrit: Subjektive Theorien - Rekonstruktion und Dialog-Konsens. In: *Siegener Periodikum für internationale empirische Literaturwissenschaft* 18 (1999), S. 138–153
- [Claren u. Sedelmaier 2012] CLAREN, Sascha; SEDELMAIER, Yvonne: Ein Kompetenzrahmenmodell für Software Engineering: Ein Schema zur Beschreibung von Kompetenzen. In: *Embedded Software Engineering Kongress 2012*, Elektronikpraxis/Vogel, 2012, S. 647–652
- [Clark et al. 2004] CLARK, Tony; EVANS, Andy; SAMMUT, Paul; WILLANS, James: *Applied Metamodelling: A Foundation for Language Driven Development*. CE-TEVA, 2004
- [Code School LLC 2016] CODE SCHOOL LLC: *Code School*. <https://www.codeschool.com/>. Version: 2016, Abruf: 25.05.2016
- [Combemale et al. 2017] COMBEMALE, Benoit; FRANCE, Robert; JÉZÉQUEL, Jean-Marc; RUMPE, Bernhard; STEEL, James; VOJTISEK, Didier: *Engineering Modeling Languages: Turning Domain Knowledge into Tools*. Boca Raton: Taylor & Francis, CRC Press, 2017 (Chapman & Hall/CRC Innovations in Software Engineering and Software Development Series)
- [Conor et al. 2014] CONOR, Linehan; KIRMAN, Ben; ROCHE, Bryan: Gamification as Behavioral Psychology. In: WALZ, Steffen P. (Hrsg.); DETERDING, Sebastian (Hrsg.): *The gameful world*. Cambridge, Mass. [u.a.]: MIT press, 2014, S. 81–105

- [Consentino et al. 2017] CONSENTINO, Valèrio; GÉRARD, Sébastien; CABOT, Jordi: A Model-based Approach to Gamify the Learning of Modeling. In: FRANCH, Xavier (Hrsg.); SNOECK, Monique (Hrsg.); GUIZZARDI, Renata (Hrsg.); JURETA, Ivan (Hrsg.): *Proceedings of the 5th Symposium on Conceptual Modeling Education and the 2nd International iStar Teaching Workshop co-located with the 36th International Conference on Conceptual Modeling (ER 2017)*, 2017, S. 15–24
- [Cooper 1988] COOPER, Harris M.: Organizing knowledge syntheses: A taxonomy of literature reviews. In: *Knowledge in Society* 1 (1988), Nr. 1, S. 104–126
- [Cooper 1998] COOPER, Harris M.: *Synthesizing research: A guide for literature reviews*. Bd. 2. 3 Aufl. Thousand Oaks, Calif.: SAGE Publications, 1998
- [Coursera Inc. 2016] COURSERA INC.: *Coursera*. <https://www.coursera.org/>. Version: 2016, Abruf: 25.05.2016
- [Crouch et al. 2007] CROUCH, Catherine H.; WATKINS, Jessica; FAGEM, Adam P.; MAZUR, Eric: Peer Instruction: Engaging Students One-on-One, All At Once. In: REDISH, Edward F. (Hrsg.); COONEY, Patrick J. (Hrsg.): *Research-Based Reform of University Physics* Bd. 1. 2007, S. 1–55
- [Csikszentmihalyi 2010] CSIKSZENTMIHALYI, Mihaly: *Flow: Das Geheimnis des Glücks*. 15. Aufl. Stuttgart: Klett-Cotta, 2010
- [Dalgarno u. Fowler 2008] DALGARNO, Mark; FOWLER, Matthew: UML vs. Domain-Specific Languages. In: *Methods & Tools - Practical knowledge for the software developer, tester and project manager* 16 (2008), Nr. 2, 2–8. <http://www.methodsandtools.com/PDF/mt200802.pdf>, Abruf: 18.04.2018
- [Danelli 2015] DANELLI, Federico: Implementing Game Design in Gamification. In: REINERS, Torsten (Hrsg.); WOOD, Lincoln C. (Hrsg.): *Gamification in Education and Business*. Cham: Springer International Publishing, 2015, S. 67–79
- [Daumiller et al. 2015] DAUMILLER, Martin; FIGAS, Paula; DRESEL, Markus: Selbstbezogene Ziele von Dozierenden: Ergebnisse einer Interviewstudie. In: *Beiträge zur Hochschulforschung* 37 (2015), Nr. 4, S. 52–63
- [Davis 2009] DAVIS, Janet: Experiences with Just-in-time Teaching in Systems and Design Courses. In: *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, ACM, 2009 (SIGCSE '09), S. 71–75



- [de Byl 2012] DE BYL, Penny: Can digital natives level-up in a gamified curriculum? In: BROWN, Mark (Hrsg.); HARTNETT, Maggie (Hrsg.); STEWART, Terry (Hrsg.): *Future Challenges, Sustainable Futures*, Massey University, 2012, S. 25–28
- [de Byl u. Hooper 2013] DE BYL, Penny; HOOPER, James: Key Attributes of Engagement in a Gamified Learning Environment. In: CARTER, Helen (Hrsg.); GOSPER, Maree (Hrsg.); HEDBERG, John (Hrsg.): *Electric dreams*, Macquarie University and Aacquarie University, 2013, S. 221–230
- [de Marcos et al. 2014] DE MARCOS, Luis; DOMÍNGUEZ, Adrián; NAVARRETE, Joseba Saenz-de; PAGÉS, Carmen: An empirical study comparing gamification and social networking on e-learning. In: *Computers & Education* 75 (2014), S. 82–91
- [de Marcos et al. 2016] DE MARCOS, Luis; GARCIA-LOPEZ, Eva; GARCIA-CABOT, Antonio: On the Effectiveness of Game-like and Social Approaches in Learning: Comparing Educational Gaming, Gamification & Social Networking. In: *Computers & Education* (2016), Nr. 95, S. 99–113
- [de Sousa Borges et al. 2014] DE SOUSA BORGES, Simone; DURELLI, Vinicius H. S.; REIS, Helena M.; ISOTANI, Seiji: A Systematic Mapping on Gamification Applied to Education. In: CHO, Yookun (Hrsg.); SHIN, Sung Y. (Hrsg.); KIM, Sangwook (Hrsg.); HUNG, Chih-Cheng (Hrsg.); HONG, Jiman (Hrsg.): *Proceedings of the 29th Annual ACM Symposium on Applied Computing 2014*, ACM, 2014, S. 216–222
- [de Troyer et al. 2017] DE TROYER, Olga; VAN BROECKHOVEN, Frederik; Vlieghe, Joachim: Creating Story-Based Serious Games Using a Controlled Natural Language Domain Specific Modeling Language. In: MA, Minhua (Hrsg.); OIKONOMOU, Andreas (Hrsg.): *Serious Games and Edutainment Applications*. Cham: Springer, 2017, S. 567–603
- [Deci u. Ryan 1985] DECI, Edward L.; RYAN, Richard M.: *Intrinsic Motivation and Self-Determination in Human Behavior*. Boston, MA: Springer US, 1985
- [Deci u. Ryan 1993] DECI, Edward L.; RYAN, Richard M.: Die Selbstbestimmungstheorie der Motivation und ihre Bedeutung für die Pädagogik. In: *Zeitschrift für Pädagogik* 39 (1993), S. 223–238
- [Deci u. Ryan 2002] DECI, Edward L.; RYAN, Richard M.: *Handbook of self-determination research*. Rochester, NY: University of Rochester Press, 2002

- [Den Haan 2009] DEN HAAN, Johan: *DSL development: 7 recommendations for Domain Specific Language design based on Domain-Driven Design*. 2009
- [Denning 1997] DENNING, Peter J.: A new social contract for research. In: *Communications of the ACM* 40 (1997), Nr. 2, S. 132–134
- [Deterding 2011a] DETERDING, Sebastian: *Meaningful Play - Getting Gamification Right*. <https://www.youtube.com/watch?v=7ZGCPap7GkY&feature=youtu.be>. Version: 2011, Abruf: 18.04.2018
- [Deterding 2011b] DETERDING, Sebastian: *A Quick Buck by Copy and Paste*. <http://gamification-research.org/2011/09/a-quick-buck-by-copy-and-paste/>. Version: 2011, Abruf: 18.04.2018
- [Deterding 2012] DETERDING, Sebastian: Gamification: designing for motivation. In: *interactions* 19 (2012), Nr. 4, 14–17. <http://dl.acm.org/citation.cfm?id=2212883>, Abruf: 28.05.2018
- [Deterding 2013a] DETERDING, Sebastian: Gameful design for learning. In: *T and D* 67 (2013), Nr. 7, 60–63. <https://www.td.org/Publications/Magazines/TD/TD-Archive/2013/07/Gameful-Design-for-Learning>, Abruf: 20.05.2015
- [Deterding 2013b] DETERDING, Sebastian: Skill Atoms as Design Lenses for User-Centered Gameful Design. In: *Conference on Human Factors in Computing Systems (CHI)'13 Workshop Proceedings*, ACM, 2013, S. 1–4
- [Deterding 2014] DETERDING, Sebastian: Eudaimonic Design, or: Six Invitations to Rethink Gamification. In: FUCHS, Mathias (Hrsg.); FIZEK, Sonia (Hrsg.); RUFFINO, Paolo (Hrsg.); SCHRAPE, Niklas (Hrsg.): *Rethinking Gamification*. meson press, 2014, S. 305–332
- [Deterding 2015] DETERDING, Sebastian: The Lens of Intrinsic Skill Atoms: A Method for Gameful Design. In: *Human-Computer Interaction* 30 (2015), Nr. 3-4, S. 294–335
- [Deterding 2016] DETERDING, Sebastian: Make-Believe in Gameful and Playful Design. In: TURNER, Phil (Hrsg.); HARVIAINEN, Tuomas J. (Hrsg.): *Digital Make-Believe*. Cham, Switzerland: Springer International Publishing, 2016, S. 101–124
- [Deterding et al. 2013] DETERDING, Sebastian; BJÖRK, Staffan L.; NACKE, Lennart E.; DIXON, Dan; LAWLEY, Elizabeth: Designing gamification: creating gameful and playful experiences. In: MACKAY, Wendy (Hrsg.); BREWSTER,

- Stephen (Hrsg.); BØDKER, Susanne (Hrsg.): *Extended Abstracts on Human Factors in Computing Systems*, ACM, 2013, S. 3263–3266
- [Deterding et al. 2011a] DETERDING, Sebastian; DIXON, Dan; KHALED, Rilla; NACKE, Lennart: From game design elements to gamefulness: defining gamification. In: LUGMAYR, Artur (Hrsg.); FRANSSILA, Heljä (Hrsg.); SAFRAN, Christian (Hrsg.); HAMMOUDA, Imed (Hrsg.): *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, 2011, S. 9–15
- [Deterding et al. 2011b] DETERDING, Sebastian; SICART, Miguel; NACKE, Lennart E.; O'HARA, Kenton; DIXON, Dan: Gamification. Using game-design elements in non-gaming contexts. In: *Conference on Human Factors in Computing Systems (CHI)'11 - Extended Abstracts*, ACM, 2011, S. 2425–2428
- [Dichev u. Dicheva 2017] DICHEV, Christo; DICHEVA, Darina: Gamifying education: What is known, what is believed and what remains uncertain: a critical review. In: *International Journal of Educational Technology in Higher Education* 14 (2017), Nr. 1, S. 1–36
- [Dichev et al. 2014] DICHEV, Christo; DICHEVA, Darina; ANGELOVA, Galia; AGRE, Gennady: From Gamification to Gameful Design and Gameful Experience in Learning. In: *Cybernetics and Information Technologies* 14 (2014), Nr. 4, 80–100. [http://www.cit.iit.bas.bg/CIT\\_2014/v14-4/7-15-CIT2014-Dichev%20\\_1\\_-m-Gotovo.pdf](http://www.cit.iit.bas.bg/CIT_2014/v14-4/7-15-CIT2014-Dichev%20_1_-m-Gotovo.pdf), Abruf: 30.01.2015
- [Dicheva u. Dichev 2016] DICHEVA, Darina; DICHEV, Christo: An Active Learning Model Employing Flipped Learning and Gamification Strategies. In: *Proceedings of first International Workshop on Intelligent Mentoring Systems*, 2016, S. 1–6
- [Dicheva et al. 2015] DICHEVA, Darina; DICHEV, Christo; AGRE, Gennady; ANGELOVA, Galia: Gamification in Education: A Systematic Mapping Study. In: *Educational Technology & Society* 18 (2015), Nr. 3, S. 1–14
- [Dicheva et al. 2014] DICHEVA, Darina; IRWIN, Keith; DICHEV, Christo; TALASILIA, Swapna: A course gamification platform supporting student motivation and engagement. In: *2014 International Conference on Web & Open Access to Learning (ICWOAL)*, IEEE, 2014, S. 1–4
- [Dignan 2011] DIGNAN, Aaron: *Game frame: Using games as a strategy for success*. New York: Free Press, 2011

- [Dijkstra 1982] DIJKSTRA, Edsger W.: *Selected writings on computing: A personal perspective*. New York: Springer-Verlag, 1982
- [Dixon 2011] DIXON, Dan: Player Types and Gamification. In: *Conference on Human Factors in Computing Systems (CHI)'11 - Workshop Gamification*, 2011, S. 1–4
- [Dominguez et al. 2013] DOMINGUEZ, Adrian; NAVARRETE, Joseba Saenz-de; MARCOS, Luis de; FERNANDEZ-SANZ, Luis; PAGES, Carmen; MARTINEZ-HERRAIZ, Jose-Javier: Gamifying learning experiences: Practical implications and outcomes. In: *Computers & Education* 63 (2013), Nr. 63, 380–392. <http://www.sciencedirect.com/science/article/pii/S0360131513000031>, Abruf: 26.02.2015
- [Doran 1981] DORAN, George T.: There's a SMART way to write management's goals and objectives. In: *Management Review* 70 (1981), Nr. 11, S. 35–36
- [Dormans 2012] DORMANS, Joris: *Engineering Emergence: Applied Theory for Game Design*. Amsterdam, Universität Amsterdam, Diss., 2012
- [Dörnyei u. Ushioda 2013] DÖRNYEI, Zoltán; USHIODA, Ema: *Teaching and researching motivation*. 2. Aufl. London: Routledge, 2013 (Applied linguistics in action)
- [Dresel 2010] DRESEL, Markus: Förderung der Lernmotivation mit attributionalem Feedback. In: SPIEL, Christine (Hrsg.); SCHOBER, Barbara (Hrsg.); WAGNER, Petra (Hrsg.); REIMANN, Ralph (Hrsg.): *Bildungspsychologie*. Göttingen: Hogrefe, 2010, S. 131–135
- [Dresel u. Lämmle 2011] DRESEL, Markus; LÄMMLE, Lena: Motivation. In: GÖTZ, Thomas (Hrsg.): *Emotion, Motivation und selbstreguliertes Lernen*. Paderborn [u.a.]: Schöningh, 2011, S. 80–141
- [Edelmann u. Wittmann 2012] EDELMANN, Walter; WITTMANN, Simone: *Lernpsychologie*. 7. Weinheim: Beltz, 2012
- [Eeles et al. 2010] EELES, Peter; CRIPPS, Peter; BOOCH, Grady: *The Process of Software Architecting*. Upper Saddle River NJ: Addison-Wesley, 2010
- [Elmes et al. 1995] ELMES, David G.; KANTOWITZ, Barry H.; ROEDIGER, Henry L.: *Research Methods in Psychology*. 5. St. Paul, Minn.: West Pub. Co., 1995

- [Erpenbeck 2014] ERPENBECK, John: Stichwort: Kompetenzen: #PIAAC. In: *Deutsches Institut für Erwachsenenbildung (DIE)* (2014), Nr. 3, 20–21. [https://www.wbv.de/journals/zeitschriften/die-zeitschrift-fuer-erwachsenenbildung/artikel/shop/detail/name/\\_/o/1/DIE1403W/nb/o/category/797.html](https://www.wbv.de/journals/zeitschriften/die-zeitschrift-fuer-erwachsenenbildung/artikel/shop/detail/name/_/o/1/DIE1403W/nb/o/category/797.html), Abruf: 15.06.2015
- [Erpenbeck u. Sauter 2007] ERPENBECK, John; SAUTER, Werner: *Kompetenzentwicklung im Netz: New Blended Learning mit Web 2.0*. Köln: Luchterhand in Wolters Kluwer Deutschland, 2007 (Personalwirtschaft: Buch)
- [Europäische Kommission 2000] EUROPÄISCHE KOMMISSION: *Memorandum über Lebenslanges Lernen: Arbeitsdokument der Dokumentationsdienststellen*. Version: 2000. [http://www.hrk.de/uploads/tx\\_szconvention/memode.pdf](http://www.hrk.de/uploads/tx_szconvention/memode.pdf), Abruf: 05.05.2015
- [Fabricatore u. López 2014] FABRICATORE, Carlo; LÓPEZ, Ximena: Using Gameplay Patterns to Gamify Learning Experiences. In: BUSCH, Carsten (Hrsg.): *Proceedings of the 8th European Conference on Game Based Learning*, Academic Conferences and Publishing International Limited, 2014, S. 110–117
- [Federation of American Scientists 1710] FEDERATION OF AMERICAN SCIENTISTS: *Summit on Educational Games: Harnessing the Power of Video Games for Learning*. Washington, 17.10.2006
- [Ferrara 2012] FERRARA, John: *Playful design: Creating game experiences in everyday interfaces*. Brooklyn, N.Y.: Rosenfeld Media, 2012
- [Ferro et al. 2013] FERRO, Lauren S.; WALZ, Steffen P.; GREUTER, Stefan: Towards personalised, gamified systems: An investigation into game design, personality and player typologies. In: *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*, ACM, 2013
- [Figas et al. 2013] FIGAS, P.; HAGEL, Georg; BARTEL, A.: The Furtherance of Motivation in the Context of Teaching Software Engineering. In: *Proceedings of Global Engineering Education Conference (IEEE EDUCON 2013)*, IEEE Computer Society, 2013, S. 1299–1304
- [Figas et al. 2015a] FIGAS, Paula; BARTEL, Alexander; HAGEL, Georg: Task-based programming learning in higher education. In: *Proceedings of IEEE Global Engineering Education Conference (EDUCON)*, IEEE, 2015, S. 648–653
- [Figas et al. 2015b] FIGAS, Paula; BARTEL, Alexander; HAGEL, Georg: Übung macht den Meister? Lernaufgabentypen im Hochschulfach Software Engineering. In: SCHMOLITZKY, Axel W. (Hrsg.); HAUPTMANN, Anna S. (Hrsg.):

- Tagungsband des 14. Workshops Software Engineering im Unterricht der Hochschulen*, 2015, S. 21–27
- [Figas u. Hagel 2017] FIGAS, Paula; HAGEL, Georg: A Task is not a Task - Empirical Results about the Quality of Instructional Tasks in Higher Education. In: *Proceedings of IEEE Global Engineering Education Conference (EDUCON)*, IEEE, 2017, S. 679–682
- [Figas et al. 2014] FIGAS, Paula; MÜLLER-AMTHOR, Martina; BARTEL, Alexander; HAGEL, Georg: Man wächst mit seinen Aufgaben. Über die kompetenzorientierte Konstruktion von Lernaufgaben in der Hochschullehre am Beispiel von Software Engineering. In: RALLE, Bernd (Hrsg.); PREDIGER, Susanne (Hrsg.); HAMMANN, Marcus (Hrsg.); ROTHGANGEL, Martin (Hrsg.): *Lernaufgaben entwickeln, bearbeiten und überprüfen*. Münster: Waxmann Verlag GmbH, 2014, S. 246–249
- [Filippou et al. 2014] FILIPPOU, Justin; CHEONG, Christopher; CHEONG, France: Improving study habits using a behaviour change framework incorporating social motivation and gamification. In: SIAU, Keng (Hrsg.); LI, Qing (Hrsg.); GUO, Xunhua (Hrsg.): *PACIS 2014 Proceedings*, 2014
- [Fink 2014] FINK, Arlene: *Conducting research literature reviews: From the internet to paper*. 4. Thousand Oaks, Calif.: SAGE Publications, 2014
- [Fischer et al. 2010] FISCHER, Christian; WINTER, Robert; WORTMANN, Felix: Gestaltungstheorie. In: *WIRTSCHAFTSINFORMATIK* 52 (2010), Nr. 6, S. 383–386
- [Fitz-Walter et al. 2011] FITZ-WALTER, Zachary; TJONDRONEGORO, Dian; WYETH, Peta: Orientation passport: using gamification to engage university students. In: *Proceedings of the 23rd Australian Computer-Human Interaction Conference*, 2011, S. 122–125
- [Flatscher 2002] FLATSCHER, Rony G.: Metamodeling in EIA/CDIF—meta-metamodel and metamodels. In: *ACM Transactions on Modeling and Computer Simulation* 12 (2002), Nr. 4, S. 322–342
- [Fleischer 2004] FLEISCHER, Rudolf: Just-in-Time: Better Teaching in Hong Kong. In: *Proceedings of the Second Teaching and Learning Symposium*, 2004, S. 1–8
- [Flick 2007] FLICK, Uwe: *Qualitative Sozialforschung: Eine Einführung*. Reinbek bei Hamburg: Rowohlt Taschenbuch Verlag, 2007 (Rororo Rowohlts Enzyklopädie)

- [Flick 2014] FLICK, Uwe: Gütekriterien qualitativer Sozialforschung. In: BLASIUS, Jörg (Hrsg.); BAUR, Nina (Hrsg.): *Handbuch Methoden der empirischen Sozialforschung*. Wiesbaden: Springer VS, 2014, S. 411–423
- [Fogg 2009] FOGG, B. J.: A Behavior Model for Persuasive Design. In: CHATTERJEE, Samir (Hrsg.); DEV, Parvati (Hrsg.): *Proceedings of the 4th International Conference on Persuasive Technology*, ACM, 2009
- [Font et al. 2013] FONT, Jose M.; MAHLMANN, Tobias; MANRIQUE, Daniel; TOGELIUS, Julian: A Card Game Description Language. In: ESPARCIA-ALCÁZAR, Anna I. (Hrsg.): *Proceedings of 16th European Conference Applications of Evolutionary Computation*, Springer, 2013, S. 254–263
- [Fowler 2003] FOWLER, Martin: *Patterns of enterprise application architecture*. Boston: Addison-Wesley, 2003 (The Addison-Wesley signature series)
- [Fowler 2010] FOWLER, Martin: *Domain Specific Languages*. New Jersey: Pearson Education (US), 2010
- [Francisco-Aparicio et al. 2013] FRANCISCO-APARICIO, Andrés; GUTIÉRREZ-VELA, Francisco L.; ISLA-MONTES, José Luis; SANCHEZ, José Luis G.: Gamification: Analysis and Application. In: PENICHER, Victor M. (Hrsg.); PEÑALVER, Antonio (Hrsg.); GALLUD, José A. (Hrsg.): *New Trends in Interaction, Virtual Reality and Modeling*. London, UK: Springer London, 2013 (Human-Computer Interaction Series), S. 113–126
- [Frank 2013] FRANK, Ulrich: Domain-Specific Modeling Languages: Requirements Analysis and Design Guidelines. In: REINHARTZ-BERGER, Iris (Hrsg.); STURM, Arnon (Hrsg.); CLARK, Tony (Hrsg.); COHEN, Sholom (Hrsg.); BETTIN, Jorn (Hrsg.): *Domain engineering*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, S. 133–157
- [Fuchs 2016] FUCHS, Markus: *Classgame*. <https://classgame.de/>. Version: 2016, Abruf: 25.05.2016
- [Gabrysiak et al. 2011] GABRYSIK, Gregor; GIESE, Holger; LÜDERS, Alexander; SEIBEL, Andreas: How Can Metamodels Be Used Flexibly? In: *International Conference on Software Engineering (ICSE) 2011 - Workshop Proceedings*, 2011
- [Gage u. Berliner 1996] GAGE, Nathaniel L.; BERLINER, David C.: *Pädagogische Psychologie*. 5., vollst. überarb. Aufl. Weinheim [u.a.]: Psychologie-Verl.-Union, Beltz, 1996

- [Gamma et al. 1995] GAMMA, Erich; HELM, Richard; JOHNSON, Ralph E.; VLIS-SIDES, John: *Design patterns: Elements of reusable object-oriented software*. Reading, Mass.: Addison-Wesley, 1995 (Addison-Wesley professional computing series)
- [Geirhos 2015] GEIRHOS, Matthias: *Entwurfsmuster: Das umfassende Handbuch*. Bonn: Rheinwerk, 2015
- [Giannetto et al. 2013] GIANNETTO, David; CHAO, Joseph; FONTANA, Anthony: Gamification in a Social Learning Environment. In: *Issues in Informing Science & Information Technology* 10 (2013), Nr. 1, 195–207. <http://search.ebscohost.com/login.aspx?direct=true&db=aci&AN=91510531&lang=de&site=ehost-live>, Abruf: 25.02.2015
- [Gibson 1979] GIBSON, James J.: *The ecological approach to visual perception*. Boston: Houghton Mifflin, 1979
- [Gist u. Mitchell 1992] GIST, Marilyn E.; MITCHELL, Terence R.: Self-Efficacy: A Theoretical Analysis of Its Determinants and Malleability. In: *Academy of Management Review* 17 (1992), Nr. 2, S. 183–211
- [Glaser u. Strauss 1967] GLASER, Barney G.; STRAUSS, Anselm L.: *The discovery of grounded theory: Strategies for qualitative research*. Chicago: Aldine Pub. Co., 1967 (Observations)
- [Göbel et al. 2014] GÖBEL, Stefan; MEHM, Florian; WENDEL, Viktor; KONERT, Johannes; HARDY, Sandro; REUTER, Christian; GUTJAHR, Michael; DUTZ, Tim: Erstellung, Steuerung und Evaluation von Serious Games. In: *Informatik-Spektrum* 37 (2014), Nr. 6, 547–557. <ftp://ftp.kom.tu-darmstadt.de/papers/GMW+14.pdf>, Abruf: 18.04.2018
- [Gonschorek u. Schneider 2010] GONSCHOREK, Gernot; SCHNEIDER, Susanne: *Einführung in die Schulpädagogik und die Unterrichtsplanung*. 7. Aufl. Donauwörth: Auer, 2010
- [González Pérez u. Henderson-Sellers 2008] GONZÁLEZ PÉREZ, César A.; HENDERSON-SELLERS, Brian: *Metamodelling for Software Engineering*. Chichester [u.a.]: John Wiley, 2008
- [Greenblat 1974] GREENBLAT, Cathy S.: Sociological theory and the 'multiple reality' game. In: *Simulation & Games* 5 (1974), Nr. 1, S. 3–21
- [Grimm 1850] GRIMM, Jacob: *Über Schule Universität Academie. Eine an der Academie der Wissenschaften am 8 November 1849 von Jacob Grimm gehaltene Vorlesung*. Berlin: Dümmler, 1850



- [Groeben 1992] GROEBEN, Norbert: Die Inhalts-Struktur-Trennung als konstantes Dialog-Konsens-Prinzip?! In: SCHEELE, Brigitte (Hrsg.): *Struktur-Lege-Verfahren als Dialog-Konsens-Methodik*. Münster: Aschendorff, 1992 (Arbeiten zur sozialwissenschaftlichen Psychologie), S. 42–49
- [Groeben u. Scheele 2010] GROEBEN, Norbert; SCHEELE, Brigitte: Das Forschungsprogramm Subjektive Theorien. In: MEY, Günter (Hrsg.); MRUCK, Katja (Hrsg.): *Handbuch qualitative Forschung in der Psychologie*. Wiesbaden: VS Verlag für Sozialwissenschaften, 2010, S. 151–165
- [Groh 2012] GROH, Fabian: Gamification: State of the Art Definition and Utilization. In: ASAJ, Naim (Hrsg.): *4th Seminar on Research Trends in Media Informatics*, Institute of Media Informatics Ulm University, 2012, S. 39–46
- [Grotton 2012] GROTTON, Angel; COOKDAYS.COM (Hrsg.): *Coffee Mug Playful Design*. <http://www.cookdays.com/utensils/coffee-mug-playful-design/>. Version: 2012, Abruf: 05.05.2016
- [Grünberger 2014] GRÜNBERGER, Nina: Gewinnt der Mensch nur da, wo er spielt? - Ein Essay über Gamification zur Bewältigung von Anforderungen der Arbeitswelt. In: ROSC, Regina (Hrsg.); WALLMANN, Margarete (Hrsg.); HACKL, Wilfried (Hrsg.): *Ästhetische Erziehung im Digitalzeitalter: Magazin erwachsenenbildung.at*. Norderstedt: Books on Demand GmbH, 2014, S. 1–9
- [Gurka 2012] GURKA, Judith S.: JiTT in CS 1 and CS 2. In: *Journal of Computing Sciences in Colleges* 28 (2012), Nr. 2, S. 81–86
- [Hagel et al. 2013] HAGEL, Georg; MOTTOK, Jürgen; MÜLLER-AMTHOR, Martina: Drei Feedback-Zyklen in der Software Engineering-Ausbildung durch erweitertes Just-in-Time-Teaching. In: SPILLNER, Andreas (Hrsg.); LICHTER, Horst (Hrsg.): *Tagungsband des 13. Workshops Software Engineering im Unterricht der Hochschulen*, 2013, S. 17–26
- [Hamari u. Koivisto 2013] HAMARI, Juho; KOIVISTO, Jonna: Social motivations to use gamification: an empirical study of gamifying exercise. In: *Proceedings of the 21st European Conference on Information Systems*, 2013, S. 1–12
- [Hamari u. Koivisto 2014] HAMARI, Juho; KOIVISTO, Jonna: Measuring flow in gamification: Dispositional Flow Scale-2. In: *Computers in Human Behavior* 40 (2014), 133–143. <http://www.sciencedirect.com/science/article/pii/S0747563214004221>, Abruf: 18.04.2018

- [Hamari et al. 2014] HAMARI, Juho; KOIVISTO, Jonna; SARSA, Harri: Does Gamification Work? A Literature Review of Empirical Studies on Gamification. In: *Proceedings of the 47th Hawaii International Conference on System Sciences (HICSS)*, IEEE, 2014, S. 3025–3034
- [Hascher 2010] HASCHER, Tina: Learning and Emotion: Perspectives for Theory and Research. In: *European Educational Research Journal* 9 (2010), Nr. 1, S. 13–28
- [Hattie et al. 2013] HATTIE, John N.; BEYWL, Wolfgang; ZIERER, Klaus: *Lernen sichtbar machen: Überarbeitete deutschsprachige Ausgabe von "Visible Learning"*. Baltmannsweiler: Schneider Hohengehren, 2013
- [Heckhausen u. Gollwitzer 1987] HECKHAUSEN, Heinz; GOLLWITZER, Peter M.: Thought contents and cognitive functioning in motivational versus volitional states of mind. In: *Motivation and Emotion* 11 (1987), Nr. 2, S. 101–120
- [Heimann u. Schütz 2017] HEIMANN, Monika; SCHÜTZ, Michael: *Wie Design wirkt: Psychologische Prinzipien erfolgreicher Gestaltung*. Bonn: Rheinwerk Verlag, 2017
- [Heinsen u. Vogt 2003] HEINSEN, Sven (Hrsg.); VOGT, Petra (Hrsg.): *Usability praktisch umsetzen: Handbuch für Software, Web, Mobile Devices und andere interaktive Produkte*. München: Hanser, 2003
- [Henricks 2010] HENRICKS, Thomas S.: Caillois's Man, Play, and Games: An Appreciation and Evaluation. In: *American Journal of Play* 3 (2010), Nr. 2, S. 157–185
- [Herrlich et al. 2014] HERRLICH, Marc; WENIG, Dirk; WALTHER-FRANKS, Benjamin; SMEDDINCK, Jan D.; MALAKA, Rainer: 'Raus aus dem Sessel' – Computerspiele für mehr Gesundheit. In: *Informatik-Spektrum* (2014), Nr. August 2014, 1–9. <http://dx.doi.org/10.1007/s00287-014-0825-1>, Abruf: 18.04.2018
- [Herzig et al. 2013] HERZIG, Philipp; JUGEL, Kay.; MOMM, Christof.; AMELING, Michael; SCHILL, Alexander: GaML - A Modeling Language for Gamification. In: *Utility and Cloud Computing (UCC), 2013 IEEE/ACM 6th International Conference on*, 2013, S. 494–499
- [Hevner u. Chatterjee 2010] HEVNER, Alan R.; CHATTERJEE, Samir: Design Science Research in Information Systems. In: HEVNER, Alan (Hrsg.); CHATTERJEE, Samir (Hrsg.): *Design Research in Information Systems*. Boston, MA: Springer Science+Business Media, LLC, 2010 (Integrated Series in Information Systems), S. 9–22

- [Hevner et al. 2004] HEVNER, Alan R.; MARCH, Salvatore T.; PARK, Jinsoo; RAM, Sudha: Design Science in Information Systems Research. In: *MIS Q* 28 (2004), Nr. 1, 75–105. <http://dl.acm.org/citation.cfm?id=2017212.2017217>, Abruf: 28.05.2018
- [Hidi u. Anderson 1992] HIDI, Suzanne; ANDERSON, Valerie: Situational interest and its impact on reading and expository writing. In: RENNINGER, Ann K. (Hrsg.); HIDI, Suzanne (Hrsg.); KRAPP, Andreas (Hrsg.): *The role of interest in learning and development*. Hillsdale, New Jersey: Erlbaum, 1992, S. 215–238
- [Hirdes et al. 2012] HIRDES, Eike M.; THILLAINATHAN, Niroshan; LEIMEISTER, Jan M.: Towards Modeling Educational Objectives in Serious Games. In: *Proceedings Pedagogically-driven Serious Games*, 2012
- [Hoblitz 2015a] HOBLITZ, Anna: Lernmotivation und Spielmotivation. In: HOBLITZ, Anna (Hrsg.): *Spielend Lernen im Flow*. Wiesbaden: Springer Fachmedien Wiesbaden, 2015, S. 87–110
- [Hoblitz 2015b] HOBLITZ, Anna (Hrsg.): *Spielend Lernen im Flow*. Wiesbaden: Springer Fachmedien Wiesbaden, 2015
- [Huhn 2004] HUHN, Gerhard: Das Flow-Erleben als Schlüssel für Lernen, Wachstum und Motivation. In: RADATZ, S. (Hrsg.); MATURANA, H.R. (Hrsg.): *Evolutionäres Management: Antworten auf die Management- und Führungsherausforderungen im 21. Jahrhundert*. Wien: Verlag Systemisches Management, 2004, S. 248–260
- [Huizinga 2013] HUIZINGA, Johan: *Rowohlts Enzyklopädie*. Bd. 435: *Homo ludens: Vom Ursprung der Kultur im Spiel*. 23. Aufl. Reinbek bei Hamburg: Rowohlt, 2013
- [Hunicke et al. 2004] HUNICKE, Robin; LEBLANC, Marc; ZUBEK, Robert: MDA: A Formal Approach to Game Design and Game Research. In: *Proceedings of the Challenges in Games AI Workshop*, 2004, S. 1–5
- [Huotari u. Hamari 2011] HUOTARI, Kai; HAMARI, Juho: Gamification from the perspective of service marketing. In: *Conference on Human Factors in Computing Systems (CHI)'11 - Workshop Gamification*, 2011, S. 1–5
- [Huotari u. Hamari 2012] HUOTARI, Kai; HAMARI, Juho: Defining Gamification: A Service Marketing Perspective. In: *Proceedings of the 16th International Academic MindTrek Conference 2012*, 2012, S. 17–22

- [Ibanez et al. 2014] IBANEZ, Maria-Blanca; DI SERIO, Angela; DELGADO KLOOS, Carlos: Gamification for Engaging Computer Science Students in Learning Activities: A Case Study. In: *IEEE Transactions on Learning Technologies* 7 (2014), Nr. 3, 291–301. <http://www.computer.org/csdl/trans/lt/2014/03/06827214-abs.html>, Abruf: 27.02.2015
- [Iivari 2007] IIVARI, Juhani: A Paradigmatic Analysis of Information Systems As a Design Science. In: *Scandinavian Journal of Information Systems* 19 (2007), Nr. 2, S. 39–64
- [Iosup u. Epema 2014] IOSUP, Alexandru; EPEMA, Dick: An experience report on using gamification in technical higher education. In: DOUGHERTY, John (Hrsg.); NAGEL, Kris (Hrsg.); DECKER, Adrienne (Hrsg.); EISELT, Kurt (Hrsg.): *Proceedings of the 45th ACM technical symposium on Computer science education*, 2014, S. 27–32
- [iversity GmbH 2016] IVERSITY GMBH: *Iversity*. <https://iversity.org/>. Version: 2016, Abruf: 25.05.2016
- [Jakubowski 2014] JAKUBOWSKI, Michal: Designing Gamified Course for Students - Framework and Examples. In: KRIZ, Willy C. (Hrsg.): *The Shift from Teaching to Learning*. Bielefeld: Bertelsmann, W, 2014, S. 248–255
- [Jang et al. 2010] JANG, Hyungshim; REEVE, Johnmarshall; DECI, Edward L.: Engaging Students in Learning Activities: It Is Not Autonomy Support or Structure but Autonomy Support and Structure. In: *Journal of Educational Psychology* 102 (2010), Nr. 3, S. 588–600
- [Jang et al. 2009] JANG, Hyungshim; REEVE, Johnmarshall; RYAN, Richard M.; KIM, Ahyoung: Can self-determination theory explain what underlies the productive, satisfying learning experiences of collectivistically oriented Korean students? In: *Journal of Educational Psychology* 101 (2009), Nr. 3, S. 644–661
- [Järvinen 2007] JÄRVINEN, Pertti: Action Research is Similar to Design Science. In: *Quality & Quantity* 41 (2007), Nr. 1, S. 37–54
- [Juul 2005] JUUL, Jasper: *Half-Real: Video Games between Real Rules and Fictional Worlds*. Cambridge, Mass.: MIT press, 2005
- [Kamph et al. 2013] KAMPH, Timo; SALDEN, Peter; SCHUPP, Sibylle; KAUTZ, Christian: Just-in-Time Teaching für Software Engineering. In: SPILLNER, Andreas (Hrsg.); LICHTER, Horst (Hrsg.): *Tagungsband des 13. Workshops Software Engineering im Unterricht der Hochschulen*, 2013, S. 9–15

- [Kankanhalli et al. 2012] KANKANHALLI, Atreyi; TAHER, Mahdiah; CAVUSOGLU, Huseyin; KIM, Seung-Hyun: Gamification: A New Paradigm for Online User Engagement. In: *Proceedings of Thirty Third International Conference on Information Systems (ICIS 2012)*, 2012
- [Kano et al. 1984] KANO, Noriaki; SERAKU, Nobuhiko; TAKAHASHI, Fumio; ICHI TSUJI, Shin: Attractive Quality and Must-Be Quality. In: *Journal of the Japanese Society for Quality Control* 14 (1984), Nr. 2, S. 147–156
- [Kapp 2012a] KAPP, Karl M.: Games, gamification, and the quest for learner engagement. In: *T+D* 66 (2012), Nr. 6, 64–68. <http://www.questia.com/magazine/1G1-292513787/games-gamification-and-the-quest-for-learner-engagement>, Abruf: 10.12.2014
- [Kapp 2012b] KAPP, Karl M.: *The gamification of learning and instruction: game-based methods and strategies for training and education*. San Francisco, CA: John Wiley & Sons and Pfeiffer, 2012
- [Kapp 2014] KAPP, Karl M.: *Gamification is about Design, Not Technology*. <http://karlkapp.com/gamification-is-about-design-not-technology/>. Version: 2014, Abruf: 25.04.2016
- [Kapp et al. 2014] KAPP, Karl M.; BLAIR, Lucas; MESCH, Rich: *The Gamification of Learning and Instruction Fieldbook: Ideas into Practice*. San Francisco: John Wiley & Sons, 2014
- [Kappen u. Nacke 2013] KAPPEN, Dennis L.; NACKE, Lennart E.: The kaleidoscope of effective gamification: deconstructing gamification in business applications. In: NACKE, Lennart E. (Hrsg.); HARRIGAN, Kevin (Hrsg.); RANDALL, Neil (Hrsg.): *Proceedings of the First International Conference on Gameful Design, Research, and Applications*, ACM, 2013 (Gamification '13), S. 119–122
- [Katz et al. 1973] KATZ, Elihu; BLUMLER, Jay G.; GUREVITCH, Michael: Uses and Gratifications Research. In: *Public Opinion Quarterly* 37 (1973), Nr. 4, S. 509–523
- [Kecher u. Salvanos 2015] KECHER, Christoph; SALVANOS, Alexander: *UML 2.5: Das umfassende Handbuch*. 5. Aufl., rev. Ausg. Bonn: Galileo Press, 2015
- [Keller 1995] KELLER, Gustav: *Motivationsstörungen im Schulalter: Formen, Ursachen, Förderung*. Donauwörth: Auer, 1995

- [Keller u. Bender 2012] KELLER, Stefan; BENDER, Ute: *Aufgabenkulturen: Fachliche Lernprozesse herausfordern, begleiten, auswerten*. Seelze: Friedrich Verlag, 2012
- [Kelly u. Tolvanen 2000] KELLY, Steven; TOLVANEN, Juha-Pekka: Visual domain-specific modelling: Benefits and experiences of using metaCASE tools. In: BEZIVIN, Jean (Hrsg.); ERNST, Joseph (Hrsg.): *Proceedings of International Workshop on Model Engineering*, 2000
- [Kelly u. Tolvanen 2008] KELLY, Steven; TOLVANEN, Juha-Pekka: *Domain-specific modeling: Enabling full code generation*. Hoboken, N.J.: John Wiley & Sons, 2008
- [Khan Academy Inc. 2016] KHAN ACADEMY INC.: *Khan Academy*. <https://de.khanacademy.org/>. Version: 2016, Abruf: 25.05.2016
- [Kim u. Lee 2013] KIM, JungTae; LEE, Won-Hyung: Dynamical model for gamification of learning (DMGL). In: *Multimedia Tools and Applications* (2013), S. 1–11
- [Kitchenham u. Charters 2007] KITCHENHAM, Barbara; CHARTERS, Stuart; SOFTWARE ENGINEERING GROUP, SCHOOL OF COMPUTER SCIENCE AND MATHEMATICS (Hrsg.): *Guidelines for performing Systematic Literature Reviews in Software Engineering: Technical Report No. EBSE-2007-01*. Version: 2007. [http://www.elsevier.com/\\_\\_data/promis\\_misc/525444systematicreviewsguide.pdf](http://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf), Abruf: 12.03.2015
- [Klabbers 2008] KLABBERS, Jan H. G.: Terminological Ambiguity. In: *Simulation & Gaming* 40 (2008), Nr. 4, S. 446–463
- [Kleppe 2009] KLEPPE, Anneke G.: *Software language engineering: Creating domain-specific languages using metamodels*. Upper Saddle River, NJ: Addison-Wesley, 2009
- [Klippert 2009] KLIPPERT, Heinz: *Teamentwicklung im Klassenraum: Übungsbausteine für den Unterricht*. Weinheim [u.a.]: Beltz, 2009 (Pädagogik-Praxis)
- [Knaving et al. 2013] KNAVING, Kristina; BOERK, Staffan; BJÖRK, Staffan: Designing for fun and play: Exploring possibilities in design for gamification. In: *Proceedings of the First International Conference on Gameful Design, Research, and Applications - Gamification*, ACM, 2013, S. 131–134

- [Knörzer 1976] KNÖRZER, Wolfgang: *Beltz Studienbuch*. Bd. 96: *Lernmotivation*. Weinheim [u.a.]: Beltz, 1976
- [Kolb 2015] KOLB, David A.: *Experiential learning: Experience as the source of learning and development*. Second edition. Upper Saddle River, New Jersey: Pearson Education, Inc, 2015
- [Kolovos et al. 2017] KOLOVOS, Dimitrios S.; GARCÍA-DOMÍNGUEZ, Antonio; ROSE, Louis M.; PAIGE, Richard F.: Eugenia: Towards disciplined and automated development of GMF-based graphical model editors. In: *Software & Systems Modeling* 16 (2017), Nr. 1, S. 229–255
- [Koster 2013] KOSTER, Raph: *Theory of fun for game design*. Sebastopol, CA: O'Reilly Media, Inc, 2013
- [Krapp 1993] KRAPP, Andreas: Die Psychologie der Lernmotivation. Perspektiven der Forschung und Probleme ihrer pädagogischen Rezeption. In: *Zeitschrift für Pädagogik* 39 (1993), Nr. 2, S. 187–206
- [Krapp 1999] KRAPP, Andreas: Interest, motivation and learning: An educational-psychological perspective. In: *European Journal of Psychology of Education* 14 (1999), Nr. 1, S. 23–40
- [Krippendorff 1980] KRIPPENDORFF, Klaus: *Content Analysis: An Introduction to its Methodology*. London, UK: SAGE Publications, 1980
- [Kruse 2014] KRUSE, Jan: *Qualitative Interviewforschung: Ein integrativer Ansatz*. Weinheim: Beltz Juventa, 2014 (Grundlagentexte Methoden)
- [Kruse et al. 2014] KRUSE, Vincent; PLICHT, Christine; SPANNAGEL, Janna; WEHRLE, Markus; SPANNAGEL, Christian: Creatures of the Night: Konzeption und Evaluation einer Gamification-Plattform im Rahmen einer Mathematikvorlesung. In: RENSING, Christoph (Hrsg.); TRAHASCH, Stephan (Hrsg.): *Proceedings of DeLFI Workshops 2014 co-located with 12th e-Learning Conference of the German Computer Society (DeLFI 2014)*, CEUR Workshop Proceedings, 2014, S. 246–253
- [Kuckartz 2012] KUCKARTZ, Udo: *Qualitative Inhaltsanalyse. Methoden, Praxis, Computerunterstützung*. Weinheim: Juventa, 2012
- [Kuhlen 1989] KUHLEN, Rainer: *Pragmatischer Mehrwert von Information. Sprachspiele mit informationswissenschaftlichen Grundbegriffen*. Version: 1989. <https://kops.uni-konstanz.de/bitstream/handle/123456789/>

- 24784/Kuhlen\_247844.pdf?sequence=2&isAllowed=y, Abruf: 22.04.2016 (Informationswissenschaft 1/89)
- [Kuhlmann u. Sauter 2008] KUHLMANN, Annette M.; SAUTER, Werner: *Innovative Lernsysteme: Kompetenzentwicklung mit Blended Learning und Social Software*. Berlin [u.a.]: Springer, 2008 (X.media.press)
- [Kultusministerkonferenz 2014] KULTUSMINISTERKONFERENZ; SEKRETARIAT DER STÄNDIGEN KONFERENZ DER KULTUSMINISTER DER LÄNDER IN DER BUNDESREPUBLIK DEUTSCHLAND (Hrsg.): *Das Bildungswesen in der Bundesrepublik Deutschland 2012/2013: Darstellung der Kompetenzen, Strukturen und bildungspolitischen Entwicklungen für den Informationsaustausch in Europa*. Version: 2014. [https://www.kmk.org/fileadmin/Dateien/pdf/Eurydice/Bildungswesen-dt-pdfs/dossier\\_de\\_ebook.pdf](https://www.kmk.org/fileadmin/Dateien/pdf/Eurydice/Bildungswesen-dt-pdfs/dossier_de_ebook.pdf), Abruf: 23.05.2016
- [Kumar u. Khurana 2012] KUMAR, B.; KHURANA, P.: Gamification in Education - Learn Computer Programming with Fun. In: *International Journal of Computers and Distributed Systems* 2 (2012), Nr. 1, S. 46–53
- [Lages Dos Santos 2016] LAGES DOS SANTOS, Adriano: *Gamification in Software Engineering Learning: A Systematic Mapping Study*. Belo Horizonte, Brazil, 2016
- [Lamnek 2010] LAMNEK, Siegfried: *Qualitative Sozialforschung: Lehrbuch*. 5. Aufl. Weinheim [u.a.]: Beltz, 2010
- [Landers 2014] LANDERS, Richard N.: Developing a Theory of Gamified Learning: Linking Serious Games and Gamification of Learning. In: *Simulation and Gaming* 45 (2014), Nr. 6, S. 752–768
- [Landers et al. 2017] LANDERS, Richard N.; ARMSTRONG, Michael B.; COLLMUS, Andrew B.: How to Use Game Elements to Enhance Learning: Applications of the Theory of Gamified Learning. In: MA, Minhua (Hrsg.); OIKONOMOU, Andreas (Hrsg.): *Serious Games and Edutainment Applications*. Cham: Springer, 2017, S. 457–483
- [Landers et al. 2015] LANDERS, Richard N.; BAUER, Kristina N.; CALLAN, Rachel C.; ARMSTRONG, Michael B.: Psychological Theory and the Gamification of Learning. In: REINERS, Torsten (Hrsg.); WOOD, Lincoln C. (Hrsg.): *Gamification in Education and Business*. Cham: Springer International Publishing, 2015, S. 165–186



- [Landers u. Landers 2014] LANDERS, Richard N.; LANDERS, Amy K.: An Empirical Test of the Theory of Gamified Learning: The Effect of Leaderboards on Time-on-Task and Academic Performance. In: *Simulation & Gaming* 45 (2014), Nr. 6, S. 769–785
- [Langner u. Mertens 2012] LANGNER, Anne-Kristin; MERTENS, Mathias: *Flow aus Spielen: optimale Erfahrungen durch Computerspiele*. Salzhemmendorf: Blumenkamp, 2012
- [Lasry 2008] LASRY, Nathaniel: Clickers or Flashcards: Is There Really a Difference? In: *The Physics Teacher* 46 (2008), Nr. 4, S. 242–245
- [Lave u. Wenger 1990] LAVE, Jean; WENGER, Etienne: *Situated learning: Legitimate peripheral participation*. Cambridge Univ. Press, 1990 (Learning in doing)
- [Lee u. Doh 2012] LEE, Haksu; DOH, Young Y.: A Study on the Relationship between Educational Achievement and Emotional Engagement in a Gameful Interface for Video Lecture Systems. In: KELLENBERGER, Patrick (Hrsg.): *2012 International Symposium on Ubiquitous Virtual Reality (ISUVR)*, IEEE Computer Society, 2012, S. 34–37
- [Lee u. Hammer 2011] LEE, Joey J.; HAMMER, Jessica: Gamification in education: What, how, why bother? In: *Academic Exchange Quarterly* 15 (2011), Nr. 2, S. 146
- [LEGO Group 2016] LEGO GROUP: *LEGO Mindstorms EV3: Learn to Program - It's Easy*. <http://www.lego.com/en-us/mindstorms/learn-to-program>. Version: 2016, Abruf: 01.05.2016
- [Lewin 1951] LEWIN, Kurt: Problems of Research in Social Psychology. In: CARTWRIGHT, Dorwin (Hrsg.): *Field Theory in Social Science*. New York, USA: Harper & Row, 1951, S. 155–169
- [Lewis 2014a] LEWIS, Chris: Gameful Patterns. In: LEWIS, Chris (Hrsg.); CHRIS LEWIS (Hrsg.): *Irresistible Apps: Motivational Design Patterns for Apps*. Berkeley: Apress, 2014, S. 33–50
- [Lewis 2014b] LEWIS, Chris: Introduction to Motivational Design. In: LEWIS, Chris (Hrsg.); CHRIS LEWIS (Hrsg.): *Irresistible Apps: Motivational Design Patterns for Apps*. Berkeley: Apress, 2014, S. 1–7

- [Liu u. Peng 2013] LIU, Pu; PENG, Zhenghong: Gamification interaction design of online education. In: *2013 2nd International Symposium on Instrumentation & Measurement, Sensor Network and Automation (IMSNA)*, IEEE, 2013, S. 95–101
- [Liu et al. 2011] LIU, Yefeng; ALEXANDROVA, Todorka; NAKAJIMA, Tatsuo: Gamifying intelligent environments. In: *Proceedings of the 2011 international ACM workshop on Ubiquitous meta user interfaces*, 2011, S. 7–12
- [Locke u. Latham 1990] LOCKE, Edwin A.; LATHAM, Gary P.: *A theory of goal setting and task performance*. Upper Saddle River: Prentice-Hall, Inc., 1990
- [Locke u. Latham 2006] LOCKE, Edwin A.; LATHAM, Gary P.: New directions in goal-setting theory. In: *Current directions in psychological science* 15 (2006), Nr. 5, S. 265–268
- [López-Fernández et al. 2016] LÓPEZ-FERNÁNDEZ, Jesús J.; GARMENDIA, Antonio; GUERRA, Esther; DE LARA, Juan: Example-Based Generation of Graphical Modelling Environments. In: WĄSOWSKI, Andrzej (Hrsg.); LÖNN, Henrik (Hrsg.): *Modelling Foundations and Applications* Bd. 9764. Cham: Springer International Publishing, 2016, S. 101–117
- [Lorenz u. Meier 2014] LORENZ, Anja; MEIER, Stefan: Digital Badges zur Dokumentation von Kompetenzen: Klassifikation und Umsetzung am Beispiel des Saxon Open Online Courses (SOOC). In: RENSING, Christoph (Hrsg.); TRAHASCH, Stephan (Hrsg.): *Proceedings of DeLFI Workshops 2014 co-located with 12th e-Learning Conference of the German Computer Society (DeLFI 2014)*, CEUR Workshop Proceedings, 2014
- [Love et al. 2008] LOVE, Nathaniel; HINRICHS, Timothy; HALEY, David; SCHKUFFA, Eric; GENESERETH, Michael; STANFORD LOGIC GROUP (Hrsg.): *General Game Playing: Game Description Language Specification: LG-2006-01*. Version: 2008. [http://logic.stanford.edu/classes/cs227/2012/readings/gdl\\_spec.pdf](http://logic.stanford.edu/classes/cs227/2012/readings/gdl_spec.pdf), Abruf: 02.01.2018
- [Maier et al. 2010] MAIER, Uwe; KLEINKNECHT, Marc; METZ, Kerstin: Ein fächerübergreifendes Kategoriensystem zur Analyse und Konstruktion von Aufgaben. In: KIPER, Hanna (Hrsg.); MEINTS, Waltraud (Hrsg.); PETERS, Sebastian (Hrsg.); SCHLUMP, Stephanie (Hrsg.); SCHMIT, Stefan (Hrsg.): *Lernaufgaben und Lernmaterialien im kompetenzorientierten Unterricht*. Stuttgart: Kohlhammer, 2010 (Schulpädagogik), S. 28–43

- [Malone u. Lepper 1987] MALONE, Thomas W.; LEPPER, Mark R.: Making Learning Fun: A Taxonomy of Intrinsic Motivations for Learning. In: SNOW, Richard E. (Hrsg.); FARR, Marshall J. (Hrsg.): *Aptitude, Learning, and Instruction*. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1987, S. 223–253
- [Mandl u. Hense 2014] MANDL, Heinz; HENSE, Ulrich: Lernen durch Spiele: Leeres Versprechen oder echtes Potenzial? In: *Wirtschaftspsychologie aktuell* (2014), Nr. 3, S. 24–28
- [Marchiori et al. 2011] MARCHIORI, Eugenio J.; DEL BLANCO, Ángel; TORRENTE, Javier; MARTINEZ-ORTIZ, Iván; FERNÁNDEZ-MANJÓN, Baltasar: A visual language for the creation of narrative educational games. In: *Journal of Visual Languages & Computing* 22 (2011), Nr. 6, S. 443–452
- [Martinez 2012] MARTINEZ, Alexandra: Using JITT in a Database Course. In: *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education (SIGCSE '12)*, ACM, 2012, S. 367–372
- [Maslow 2013] MASLOW, Abraham H.: *Theory of human motivation*. Radford, US: Wilder Publications, 2013
- [Mayring 2002] MAYRING, Philipp: *Einführung in die qualitative Sozialforschung: Eine Anleitung zu qualitativem Denken*. 5. Aufl. Weinheim: Beltz, 2002
- [Mayring 2010] MAYRING, Philipp: Qualitative Inhaltsanalyse. In: MEY, Günter (Hrsg.); MRUCK, Katja (Hrsg.): *Handbuch qualitative Forschung in der Psychologie*. Wiesbaden: VS Verlag für Sozialwissenschaften, 2010, S. 601–613
- [Mayring 2015] MAYRING, Philipp: *Qualitative Inhaltsanalyse: Grundlagen und Techniken*. 12. Aufl. Weinheim: Beltz, 2015
- [Mayring u. Fenzl 2014] MAYRING, Philipp; FENZL, Thomas: Qualitative Inhaltsanalyse. In: BLASIUS, Jörg (Hrsg.); BAUR, Nina (Hrsg.): *Handbuch Methoden der empirischen Sozialforschung*. Wiesbaden: Springer VS, 2014, S. 543–556
- [Mazarakis 2017] MAZARAKIS, Athanasios: Gamification: Eine experimentelle Untersuchung der Spielelemente Abzeichen und Story. In: BURGHARDT, Manuel (Hrsg.); WIMMER, Raphael (Hrsg.); WOLFF, Christian (Hrsg.); WOMSER-HACKER, Christa (Hrsg.): *Mensch und Computer 2017 - Tagungsband*, Gesellschaft für Informatik e.V., 2017, S. 3–14

- [Mazur 2006] MAZUR, James E.: *Lernen und Verhalten*. 6., aktualisierte Aufl. München [u.a.]: Pearson Studium, 2006
- [McDaniel et al. 2012] MCDANIEL, Rudy; LINDGREN, Robb; FRISKICS, Jon: Using badges for shaping interactions in online learning environments. In: *2012 IEEE International Professional Communication Conference (IPCC 2012)*, IEEE, 2012, S. 55–76
- [McDonald et al. 2007] McDONALD, Marc; MUSSON, Robert; SMITH, Ross: *Practical Guide to Defect Prevention*. Redmond, Wash.: Microsoft Press, 2007
- [McGonigal 2011] MCGONIGAL, Jane: *Reality is broken: Why games make us better and how they can change the world*. New York: Penguin Press, 2011
- [McGonigal 2012] MCGONIGAL, Jane: *Besser als die Wirklichkeit! Warum wir von Computerspielen profitieren und wie sie die Welt verändern*. München: Heyne, 2012
- [Mehm et al. 2013] MEHM, Florian; REUTER, Christian; GÖBEL, Stefan: Authoring of Serious Games for Education. In: BREDL, Klaus (Hrsg.); BÖSCHE, Wolfgang (Hrsg.): *Serious games and virtual worlds in education, professional development, and healthcare*. Hershey, PA: Information Science Reference, 2013 (Premier Reference Source), S. 60–73
- [Mehm et al. 2012] MEHM, Florian; REUTER, Christian; GÖBEL, Stefan; STEINMETZ, Ralf: Future trends in game authoring tools. In: HERRLICH, Marc (Hrsg.); MALAKA, Rainer (Hrsg.); MASUCH, Maic (Hrsg.): *Proceedings of the 11th international conference on Entertainment Computing*, Springer, 2012 (Lecture Notes in Computer Science), S. 536–541
- [Mekler et al. 2013a] MEKLER, Elisa D.; BRÜHLMANN, Florian; OPWIS, Klaus; TUCH, Alexandre N.: Disassembling gamification: the effects of points and meaning on user motivation and performance. In: MACKAY, Wendy (Hrsg.); BREWSTER, Stephen (Hrsg.); BØDKER, Susanne (Hrsg.): *Extended Abstracts on Human Factors in Computing Systems*, ACM, 2013, S. 1137–1142
- [Mekler et al. 2013b] MEKLER, Elisa D.; BRÜHLMANN, Florian; OPWIS, Klaus; TUCH, Alexandre N.: Do points, levels and leaderboards harm intrinsic motivation? In: NACKE, Lennart E. (Hrsg.); HARRIGAN, Kevin (Hrsg.); RANDALL, Neil (Hrsg.): *Proceedings of the First International Conference on Gameful Design, Research, and Applications*, ACM, 2013 (Gamification '13), S. 66–73

- [Mekler et al. 2017] MEKLER, Elisa D.; BRÜHLMANN, Florian; TUCH, Alexandre N.; OPWIS, Klaus: Towards understanding the effects of individual gamification elements on intrinsic motivation and performance. In: *Computers in Human Behavior* 71 (2017), S. 525–534
- [Meyer 2007] MEYER, Bertrand: *Object oriented software construction*. 2. Aufl. Upper Saddle River, NJ: Prentice Hall, 2007
- [Meyer 2016] MEYER, Hilbert: *Unterrichtsmethoden*. 16. Aufl. Frankfurt am Main: Cornelsen, 2016
- [Michael u. Chen 2005] MICHAEL, David; CHEN, Sande: *Serious Games: Games That Educate, Train, and Info*. Boston, Mass.: Thomson Course Technology, 2005
- [Miede 2012] MIEDE, André: *A Classic Thesis Style*. <http://www.ctan.org/tex-archive/macros/latex/contrib/classicthesis/>. Version: 2012, Abruf: 26.11.2015
- [Mietzel 2007] MIETZEL, Gerd: *Pädagogische Psychologie des Lernens und Lehrens*. 8., überarb. und erw. Aufl. Göttingen [u.a.]: Hogrefe, 2007
- [Miller 1956] MILLER, George A.: The magical number seven, plus or minus two: Some limits on our capacity for processing information. In: *Psychological Review* 63 (1956), Nr. 2, S. 81–97
- [Millett u. Tune 2015] MILLETT, Scott; TUNE, Nick: *Patterns, Principles, and Practices of Domain-Driven Design*. Indianapolis: John Wiley & Sons, 2015
- [Möller 2006] MÖLLER, Christine: Die curriculare Didaktik: Oder: Der lernzielorientierte Ansatz. In: GUDJONS, Herbert (Hrsg.); WINKEL, Rainer (Hrsg.): *Didaktische Theorien*. Hamburg: Bergmann + Helbig, 2006, S. 75–92
- [Moody 2009] MOODY, Daniel: The “Physics” of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. In: *IEEE Transactions on Software Engineering* 35 (2009), Nr. 6, S. 756–779
- [Morford et al. 2014] MORFORD, Zachary H.; WITTS, Benjamin N.; KILLINGSWORTH, Kenneth J.; ALAVOSIUS, Mark P.: Gamification: The Intersection between Behavior Analysis and Game Design Technologies. In: *The Behavior Analyst* 37 (2014), Nr. 1, S. 25–40
- [Morris 1938] MORRIS, Charles W.: *International encyclopedia of unified science Foundations of the unity of science*. Bd. 2: *Foundations of the theory of signs*. 14. Chicago [u.a.]: Chicago University Press, 1938

- [Morschheuser et al. 2017] MORSCHHEUSER, Benedikt; WERDER, Karl; HAMARI, Juho; ABE, Julian: How to gamify? A method for designing gamification. In: *Proceedings of the 50th Hawaii International Conference on System Sciences (HICSS)*, IEEE, 2017, S. 1298–1307
- [Morse 1994] MORSE, Janice M.: Designing Funded Qualitative Research. In: DENZIN, Norman K. (Hrsg.); LINCOLN, Yvonna S. (Hrsg.): *Handbook of Qualitative Research*. Thousand Oaks: SAGE Publications, 1994, S. 220–235
- [Müller 2014] MÜLLER, Jens: Gamification - Weltrettung durch Zocken? In: KAISER, Markus (Hrsg.): *Ringvorlesung Games*. München: Hooffacker, 2014, S. 61–79
- [Nacke u. Deterding 2017] NACKE, Lennart E.; DETERDING, Sebastian: The maturing of gamification research. In: *Computers in Human Behavior* 71 (2017), S. 450–454
- [Nah et al. 2014a] NAH, Fiona Fui-Hoon; ESCHENBRENNER, Brenda; ZENG, Qing; TELAPROLU, Venkata R.; SEPEHR, Sepandar: Flow in gaming: literature synthesis and framework development. In: *International Journal of Information Systems and Management* 1 (2014), Nr. 1, S. 83–124
- [Nah et al. 2013] NAH, Fiona Fui-Hoon; TELAPROLU, Venkata R.; RALLAPALLI, Shashank; VENKATA, Pavani R.: Gamification of Education Using Computer Games. In: YAMAMOTO, Sakae (Hrsg.): *Human Interface and the Management of Information. Information and Interaction for Learning, Culture, Collaboration and Business* Bd. 8018. Springer Berlin Heidelberg, 2013, S. 99–107
- [Nah et al. 2014b] NAH, Fiona Fui-Hoon; ZENG, Qing; TELAPROLU, Venkata R.; AYYAPPA, Abhishek P.; ESCHENBRENNER, Brenda: Gamification of Education: A Review of Literature. In: NAH, Fiona Fui-Hoon (Hrsg.): *Proceedings of HCI in Business: First International Conference, HCIB 2014, Held as Part of HCI International 2014*, Springer, 2014, S. 401–409
- [Navarro 2006] NAVARRO, Emily: *SimSE: A Software Engineering Simulation Environment for Software Process Education*. Irvine, University of California, Irvine, Diss., 2006
- [Nelson 2006] NELSON, Graham: *Natural language, semantic analysis and interactive fiction*. Version: 2006. <http://inform7.com/learn/documents/WhitePaper.pdf>, Abruf: 08.01.2018

- [Ng 2013] NG, Pan-Wei: Making Software Engineering Education Structured, Relevant and Engaging through Gaming and Simulation. In: *Journal of Communication and Computer* 10 (2013), Nr. 11, S. 1365–1373
- [Nicholson 2012] NICHOLSON, Scott: A user-centered theoretical framework for meaningful gamification. In: *Games+Learning+Society* 8 (2012)
- [Nicholson 2015] NICHOLSON, Scott: A RECIPE for Meaningful Gamification. In: REINERS, Torsten (Hrsg.); WOOD, Lincoln C. (Hrsg.): *Gamification in Education and Business*. Cham: Springer International Publishing, 2015, S. 1–20
- [Nielsen u. Landauer 1993] NIELSEN, Jakob; LANDAUER, Thomas K.: A mathematical model of the finding of usability problems. In: ARNOLD, Bert (Hrsg.); VAN DER VEER, Gerrit (Hrsg.); WHITE, Ted (Hrsg.): *Proceedings of the Conference on Human Factors in Computing Systems (CHI)'93*, ACM, 1993, S. 206–213
- [Niman 2014] NIMAN, Neil B.: *The Gamification of Higher Education: Developing a Game-Based Business Strategy in a Disrupted Marketplace*. London, UK: Palgrave Macmillan, 2014
- [Norman 1995] NORMAN, Donald A.: *The psychology of everyday things*. 16. Aufl. London: BasicBooks, 1995
- [Norman 2002] NORMAN, Donald A.: *The design of everyday things*. New York, USA: Basic Books, 2002
- [Novak 2010] NOVAK, Gregor: An Introduction to Just-in-Time Teaching (JiTT). In: SIMKINS, Scott (Hrsg.); MAIER, Mark H. (Hrsg.): *Just-in-Time Teaching*. Sterling, Virginia: Stylus Publishing, 2010, S. 3–23
- [Novak et al. 1999] NOVAK, Gregor M.; PATTERSON, Evelyn T.; GAVRIN, Andrew D.; CHRISTIAN, Wolfgang: *Just-in-time teaching: Blending active learning with web technology*. Upper Saddle River, NJ: Prentice Hall, 1999
- [Obeo GmbH 2017a] OBEO GMBH: *Eclipse Acceleo*. <http://wiki.eclipse.org/Acceleo>. Version: 2017, Abruf: 10.10.2017
- [Obeo GmbH 2017b] OBEO GMBH: *Sirius Documentation*. <https://www.eclipse.org/sirius/doc/>. Version: 2017, Abruf: 18.12.2017
- [O'Donovan et al. 2013] O'DONOVAN, Siobhan; GAIN, James; MARAIS, Patrick: A case study in the gamification of a university-level games development

- course. In: McNEILL, John (Hrsg.); BRADSHAW, Karen (Hrsg.); MACHANICK, Philip (Hrsg.); TSIETSI, Mosiuoa (Hrsg.): *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*, ACM, 2013 (ACM international conference series), S. 242–251
- [Okoli u. Schabram 2010] OKOLI, Chitu; SCHABRAM, Kira: A guide to conducting a systematic literature review of information systems research. In: *Sprouts: Working Papers on Information* 10 (2010), Nr. 26. [http://aisel.aisnet.org/sprouts\\_all/348/](http://aisel.aisnet.org/sprouts_all/348/), Abruf: 28.05.2018
- [OMG 2011] OMG: *OMG Unified Modeling Language (OMG UML) Infrastructure*. <http://www.omg.org/spec/UML/2.4.1/Infrastructure/PDF/changebarred/>. Version: 2.4.1, 2011, Abruf: 04.02.2016
- [OMG 2014] OMG: *OMG Meta Object Facility (MOF) Core Specification*. <http://www.omg.org/spec/MOF/ISO/19508/PDF/>. Version: 2.4.2, 2014, Abruf: 04.02.2016
- [O'Neill et al. 2006] O'NEILL, Jan; CONZEMIUS, Anne; COMMODORE, Carol; PULSFUS, Carol: *The Power of SMART goals: Using goals to improve student learning*. Bloomington, Ind.: Solution Tree Press, 2006
- [Oracle Corporation 2011] ORACLE CORPORATION: *Core J2EE Patterns - Data Access Object*. <http://www.oracle.com/technetwork/java/dataaccessobject-138824.html>. Version: 2011, Abruf: 29.01.2018
- [Paharia 2014] PAHARIA, Rajat: *A New Day for Gamification, or Is It?* [http://www.huffingtonpost.com/rajat-paharia/a-new-day-for-gamificatio\\_b\\_5194055.html](http://www.huffingtonpost.com/rajat-paharia/a-new-day-for-gamificatio_b_5194055.html). Version: 2014, Abruf: 05.06.2015
- [Pargas 2006] PARGAS, Roy P.: Reducing lecture and increasing student activity in large computer science courses. In: DAVOLI, Renzo (Hrsg.): *Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education (ITiCSE)*, ACM, 2006, S. 3–7
- [Patterson 1999] PATTERSON, Evelyn T.: *JiTt Puzzle Pot Physics*. <http://webphysics.iupui.edu/jittworkshop/251Sp99PZL1.html>. Version: 1999, Abruf: 06.04.2016
- [Patton 2006] PATTON, Michael Q.: Foreword: Trends and issues as context. In: *Research In The Schools* 13 (2006), Nr. 1, S. i–ii
- [Pavlas et al. 2010] PAVLAS, Davin; HEYNE, Kyle; BEDWELL, Wendy; LAZZARA, Elizabeth; SALAS, Eduardo: *Game-based Learning: The Impact of Flow State*



- and Videogame Self-efficacy. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 54 (2010), Nr. 28, S. 2398–2402
- [Pedreira et al. 2015] PEDREIRA, Oscar; GARCIA, Felix; BRISABOA, Nieves; PIATTINI, Mario: Gamification in software engineering – A systematic mapping. In: *Information and Software Technology* 57 (2015), S. 157–168
- [Peppers et al. 2007] PEFTERS, Ken; TUUNANEN, Tuure; ROTHENBERGER, Marcus A.; CHATTERJEE, Samir: A Design Science Research Methodology for Information Systems Research. In: *Journal of Management Information Systems* 24 (2007), Nr. 3, S. 45–77
- [Pekrun 2011] PEKRUN, Reinhard: Emotionen, Motivation, Selbstregulation: Gemeinsame Prinzipien und offene Fragen. In: GÖTZ, Thomas (Hrsg.): *Emotion, Motivation und selbstreguliertes Lernen*. Paderborn [u.a.]: Schöningh, 2011, S. 186–205
- [Petersen et al. 2008] PETERSEN, Kai; FELDT, Robert; MUJTABA, Shahid; MATTSOHN, Michael: Systematic mapping studies in software engineering. In: *Proceedings of the 12th international Conference on Evaluation and Assessment in Software Engineering*, 2008, S. 68–77
- [Pietrek u. Trompeter 2007] PIETREK, Georg; TROMPETER, Jens: *Modellgetriebene Softwareentwicklung: MDA und MDSD in der Praxis*. Frankfurt: Entwickler.press, 2007
- [Prensky 2001] PRENSKY, Marc: *Types of Learning and Possible Game Styles: Subkapitel entnommen aus: Digital Game-Based Learning*. New York, USA: McGraw-Hill Publishing Company Ltd., 2001
- [Przybylski et al. 2010] PRZYBYLSKI, Andrew K.; RIGBY, Scott C.; RYAN, Richard M.: A motivational model of video game engagement. In: *Review of General Psychology* 14 (2010), Nr. 2, S. 154–166
- [Raffini 1996] RAFFINI, James P.: *150 ways to increase intrinsic motivation in the classroom*. Boston: Allyn and Bacon, 1996
- [Ramirez u. Squire 2014] RAMIREZ, Dennis; SQUIRE, Kurt: Gamification and Learning. In: WALZ, Steffen P. (Hrsg.); DETERDING, Sebastian (Hrsg.): *The gameful world*. Cambridge, Mass. [u.a.]: MIT press, 2014, S. 629–652
- [Randolph 2009] RANDOLPH, Justus J.: A Guide to Writing the Dissertation Literature Review. In: *Practical Assessment, Research & Evaluation* 14 (2009), Nr. 13, 1–13. <http://pareonline.net/pdf/v14n13.pdf>, Abruf: 22.01.2015

- [Reeve 1996] REEVE, Johnmarshall: *Motivating others: Nurturing inner motivational resources*. Boston: Allyn and Bacon, 1996
- [Rehfeld 2014] REHFELD, Gunther: *Game Design und Produktion: Grundlagen, Anwendungen und Beispiele*. München: Hanser, 2014
- [Reich 2012] REICH, Kersten: *Konstruktivistische Didaktik: Das Lehr- und Studienbuch mit Online-Methodenpool*. 5 Aufl. Weinheim [u.a.]: Beltz, 2012 (Beltz Pädagogik)
- [Reid et al. 2015] REID, Alan J.; PASTER, Denise; ABRAMOVICH, Samuel: Digital badges in undergraduate composition courses: effects on intrinsic motivation. In: *Journal of Computers in Education* 2 (2015), Nr. 4, S. 377–398
- [Renken et al. 2016] RENKEN, Maggie; PEFFER, Melanie; OTREL-CASS, Kathrin; GIRAULT, Isabelle; CHIOCARRIELLO, Augusto: Scaffolding Science Learning: Promoting Disciplinary Knowledge, Science Process Skills, and Epistemic Processes. In: RENKEN, Maggie (Hrsg.); PEFFER, Melanie (Hrsg.); OTREL-CASS, Kathrin (Hrsg.); GIRAULT, Isabelle (Hrsg.); CHIOCARRIELLO, Augusto (Hrsg.): *Simulations as Scaffolds in Science Education*. Cham: Springer International Publishing, 2016 (SpringerBriefs in Educational Communications and Technology), S. 23–28
- [Resnick et al. 2009] RESNICK, Mitchel; SILVERMAN, Brian; KAFAI, Yasmin; MALONEY, John; MONROY-HERNÁNDEZ, Andrés; RUSK, Natalie; EASTMOND, Evelyn; BRENNAN, Karen; MILLNER, Amon; ROSENBAUM, Eric; SILVER, Jay: Scratch: programming for all. In: *Communications of the ACM* 52 (2009), Nr. 11, S. 60
- [Rey 2009] REY, Günter D.: *E-Learning: Theorien, Gestaltungsempfehlungen und Forschung*. Bern: Huber, 2009 (Psychologie Lehrbuch)
- [Rheinberg 1986] RHEINBERG, Falko: Lernmotivation. In: FRICKE, Reiner (Hrsg.); SARGES, Werner (Hrsg.): *Psychologie für die Erwachsenenbildung/Weiterbildung*. Göttingen: C.J. Hogrefe, 1986, S. 360–365
- [Rheinberg 1989] RHEINBERG, Falko: *Motivationsforschung*. Bd. 11: *Zweck und Tätigkeit: Motivationspsychologische Analysen zur Handlungsveranlassung*. Göttingen: C.J. Hogrefe, 1989
- [Rheinberg 2008] RHEINBERG, Falko: *Kohlhammer-Urban-Taschenbücher*. Bd. Bd. 555: *Motivation*. 7. aktualisierte Aufl. Stuttgart: Kohlhammer, 2008

- [Rheinberg u. Fries 1998] RHEINBERG, Falko; FRIES, Stefan: Förderung der Lernmotivation: Ansatzpunkte, Strategien und Effekte. In: *Psychologie in Erziehung und Unterricht* 45 (1998), Nr. 3, 168–184. <http://ub-madoc.bib.uni-mannheim.de/13638/>
- [Rheinberg u. Krug 2005] RHEINBERG, Falko; KRUG, Siegbert: *Motivationsförderung im Schulalltag: Psychologische Grundlagen und praktische Durchführung*. 2. Aufl. Bern: Hogrefe, 2005
- [Rheinberg et al. 2000] RHEINBERG, Falko; VOLLMEYER, Regina; BURNS, Bruce D.: Motivation and self-regulated learning. In: HECKHAUSEN, Jutta (Hrsg.): *Motivational psychology of human development*. Amsterdam: Elsevier, 2000 (Advances in psychology), S. 81–108
- [Richter et al. 2015] RICHTER, Ganit; RABAN, Daphne R.; RAFAELI, Sheizaf: Studying Gamification: The Effect of Rewards and Incentives on Motivation. In: REINERS, Torsten (Hrsg.); WOOD, Lincoln C. (Hrsg.): *Gamification in Education and Business*. Cham: Springer International Publishing, 2015, S. 21–46
- [Rickinson u. May 2009] RICKINSON, Mark; MAY, Helen; THE HIGHER EDUCATION ACADEMY (Hrsg.): *A comparative study of methodological approaches to reviewing literature*. Version: 2009. <https://www.heacademy.ac.uk/sites/default/files/resources/Comparativestudy.pdf>, Abruf: 22.07.2015
- [Riegler 2012] RIEGLER, Peter: Just in Time Teaching. Wer liest und wer lehrt an der Hochschule? In: WALDHERR, Franz (Hrsg.); WALTER, Claudia (Hrsg.): *Forum der Lehre 2012 - Wissen, Können, verantwortlich Handeln*, 2012, S. 89–96
- [Rigby 2014] RIGBY, Scott: Gamification and Motivation. In: WALZ, Steffen P. (Hrsg.); DETERDING, Sebastian (Hrsg.): *The gameful world*. Cambridge, Mass. [u.a.]: MIT press, 2014, S. 113–137
- [Rigby u. Ryan 2011] RIGBY, Scott; RYAN, Richard M.: *Glued to Games: How Video Games Draw Us In and Hold Us Spellbound: How Video Games Draw Us In and Hold Us Spellbound*. ABC-CLIO, 2011
- [Ritsert 1975] RITSERT, Jürgen: *Inhaltsanalyse und Ideologiekritik: Ein Versuch über kritische Sozialforschung*. Bd. 4001. 2. Aufl. Frankfurt am Main: Athenäum-Fischer, 1975
- [Robertson 2010] ROBERTSON, Margaret: *Can't play, won't play*. <http://hideandseek.net/2010/10/06/cant-play-wont-play/>. Version: 2010, Abruf: 12.03.2015

- [Roderus 2015] RODERUS, Simon: Ein Kurskonzept zur Förderung des kontinuierlichen Lernens durch den Einsatz von Gamification. In: ZENTRUM FÜR HOCHSCHULDIDAKTIK (Hrsg.): *Tagungsband zum 2. HDMINT Symposium 2015*, 2015, S. 81–86
- [Rojas et al. 2013] ROJAS, David; KAPRALOS, Bill; DUBROWSKI, Adam: The Missing Piece in the Gamification Puzzle. In: NACKE, Lennart E. (Hrsg.); HARRIGAN, Kevin (Hrsg.); RANDALL, Neil (Hrsg.): *Proceedings of the First International Conference on Gameful Design, Research, and Applications*, ACM, 2013 (Gamification '13), S. 135–138
- [Rose u. Meyer 2002] ROSE, David H.; MEYER, Anne: *Teaching every student in the digital age: Universal design for learning*. Alexandria, Va: Association for Supervision and Curriculum Development, 2002
- [Rossi 2009] ROSSI, Matti: Action Design Research - An Integrative Research Method for Studying Design. In: JARKE, Matthias (Hrsg.); LYYTINEN, Kalle (Hrsg.); MYLOPOULOS, John (Hrsg.): *Perspectives Workshop: Science of Design: High-Impact Requirements for Software-Intensive Systems*, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2009 (Dagstuhl Seminar Proceedings)
- [Roth 1983] ROTH, Heinrich: *Pädagogische Psychologie des Lehrens und Lernens*. 16. Aufl. Hannover: Schroedel Schulbuchverlag, 1983
- [Roth 2015] ROTH, Jürgen: Lernpfade - Definition, Gestaltungskriterien und Unterrichtseinsatz. In: ROTH, Jürgen (Hrsg.); SÜSS-STEPANCIK, Evelyn (Hrsg.); WIESNER, Heike (Hrsg.): *Medienvielfalt im Mathematikunterricht*. Wiesbaden: Springer Fachmedien, 2015, S. 3–25
- [Rousseau et al. 2008] ROUSSEAU, Denise M.; MANNING, Joshua; DENYER, David: Evidence in Management and Organizational Science: Assembling the Field's Full Weight of Scientific Knowledge through Syntheses. In: *SSRN Electronic Journal* (2008), 1–78. [http://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=1309606](http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1309606), Abruf: 26.01.2015
- [Rowley u. Slack 2004] ROWLEY, Jennifer; SLACK, Francis: Conducting a literature review. In: *Management Research News* 27 (2004), Nr. 6, S. 31–39
- [Rudolph 2013] RUDOLPH, Udo: *Motivationspsychologie kompakt*. 3. Aufl. Weinheim [u.a.]: Beltz, 2013

- [Rughinis 2013a] RUGHINIS, Razvan: Flexible Gamification in a Social Learning Situation: Insights from a collaborative review exercise. In: RUMMEL, Nikol (Hrsg.); KAPUR, Manu (Hrsg.); NATHAN, Mitchell (Hrsg.); PUNTAMBEKAR, Sadhana (Hrsg.): *10th International Conference on Computer Supported Collaborative Learning CSCL 2013 (Volume 2)*, International Society of the Learning Sciences (ISLS), 2013, S. 137–140
- [Rughinis 2013b] RUGHINIS, Razvan: Gamification for productive interaction: Reading and working with the gamification debate in education. In: ROCHA, Álvaro (Hrsg.): *Proceedings of the 8th Iberian Conference on Information Systems and Technologies (CISTI 2013)*, IEEE, 2013, S. 521–525
- [Rupp 2009] RUPP, Chris: *Requirements-Engineering und -Management: Professionelle, iterative Anforderungsanalyse für die Praxis*. 5. aktualisierte und erw. Aufl. München: Hanser, 2009
- [Ryan 2001] RYAN, Marie-Laure: *Narrative as virtual reality: Immersion and interactivity in literature and electronic media*. Baltimore: Johns Hopkins University Press, 2001 (Parallax)
- [Ryan u. Deci 2000a] RYAN, Richard M.; DECI, Edward L.: Intrinsic and extrinsic motivations: Classic definitions and new directions. In: *Contemporary educational psychology* 25 (2000), Nr. 1, 54–67. <http://www.sciencedirect.com/science/article/pii/S0361476X99910202>, Abruf: 28.05.2018
- [Ryan u. Deci 2000b] RYAN, Richard M.; DECI, Edward L.: Self-Determination Theory and the Facilitation of Intrinsic Motivation, Social Development, and Well-Being. In: *American Psychologist* 55 (2000), Nr. 1, S. 68–78
- [Ryzac 2016] RYZAC, Inc.: *Codecademy*. <https://www.codecademy.com/>. Version: 2016, Abruf: 25.05.2016
- [Sailer 2016] SAILER, Michael: *Die Wirkung von Gamification auf Motivation und Leistung: Empirische Studien im Kontext manueller Arbeitsprozesse*. Wiesbaden: Springer, 2016
- [Salen u. Zimmerman 2004] SALEN, Katie; ZIMMERMAN, Eric: *Rules of play: Game design fundamentals*. Cambridge, Mass.: MIT press, 2004
- [Saliger 2018] SALIGER, Sebastian: *Konzeption und Implementierung einer spielifizierten mobilen App als Teil einer Lernplattform: Bachelor Thesis - Hochschule für angewandte Wissenschaften Kempten*. 2018

- [Sangkyun u. Ko 2013] SANGKYUN, Kim; KO, Franz I. S.: Toward Gamified Classroom: Classification of Engineering Students Based on The Bartle's Player Types Model. In: *International Journal of Digital Content Technology & its Applications* 7 (2013), Nr. 2, 25–31. <http://web.b.ebscohost.com/ehost/detail/detail?sid=e4832de6-6421-45ef-beee-f398555471fa%40sessionmgr113&vid=o&hid=124&bdata=JmxhbmcyZGUmc2loZT1laG9zdC1saXZl#db=aci&AN=98893771>, Abruf: 25.02.2015
- [Sauerwein 2000] SAUERWEIN, Elmar: *Das Kano-Modell der Kundenzufriedenheit*. Wiesbaden: Deutscher Universitätsverlag, 2000
- [Scheele u. Groeben 1984] SCHEELE, Brigitte; GROEBEN, Norbert: *Die Heidelberger Struktur-Lege-Technik (SLT): E. Dialog-Konsens-Methode zur Erhebung subjektiver Theorien mittlerer Reichweite*. Weinheim [u.a.]: Beltz, 1984
- [Scheele u. Groeben 1988] SCHEELE, Brigitte; GROEBEN, Norbert: *Dialog-Konsens-Methoden zur Rekonstruktion subjektiver Theorien: D. Heidelberger Struktur-Lege-Technik (SLT), konsensuale Ziel-Mittel- Argumentation u. kommunikative Flußdiagramm-Beschreibung von Handlungen*. Tübingen: Francke, 1988
- [Scheele u. Groeben 2010] SCHEELE, Brigitte; GROEBEN, Norbert: Dialog-Konsens-Methoden. In: MEY, Günter (Hrsg.); MRUCK, Katja (Hrsg.): *Handbuch qualitative Forschung in der Psychologie*. Wiesbaden: VS Verlag für Sozialwissenschaften, 2010, S. 506–523
- [Schell 2015] SCHELL, Jesse: *The Art of Game Design: A Book of Lenses*. 2. Aufl. Boca Raton: CRC Press, 2015
- [Schiefele 1996] SCHIEFELE, Ulrich: *Motivation und Lernen mit Texten*. Göttingen [u.a.]: Hogrefe, 1996
- [Schlagenhauser u. Amberg 2014] SCHLAGENHAUSER, Christian; AMBERG, Michael: Psychology Theories In Gamification: A Review of Information Systems Literature. In: HINDI, Nitham M. (Hrsg.); GHONEIM, Ahmad (Hrsg.); THEMISTOCLEOUS, Marinos (Hrsg.); VISCUSI, Gianluigi (Hrsg.): *Proceedings of the European, Mediterranean & Middle Eastern Conference on Information Systems (EMCIS)*, 2014, S. 1–12
- [Schlagenhauser u. Amberg 2015] SCHLAGENHAUSER, Christian; AMBERG, Michael: A Descriptive Literature Review and Classification Framework for

- Gamification in Information Systems. In: *Proceedings Twenty-Third European Conference on Information Systems (ECIS)*, 2015, S. 1–15
- [Schneider u. Kreuz 1979] SCHNEIDER, Klaus; KREUZ, Andreas: Die Effekte unterschiedlicher Anstrengung auf die Mengen und Güte Leistung bei einer einfachen und schweren Zahlensymbolaufgabe. In: *Psychologie und Praxis* 23 (1979), S. 34–42
- [Schöbel u. Söllner 2016] SCHÖBEL, Sofia; SÖLLNER, Matthias: How to Gamify Information Systems - Adapting Gamification to Individual User Preferences. In: *Proceedings of European Conference on Information Systems (ECIS)*, 2016, S. 1–12
- [Schöbel et al. 2016] SCHÖBEL, Sofia; SÖLLNER, Matthias; LEIMEISTER, Jan M.: The Agony of Choice – Analyzing User Preferences regarding Gamification Elements in Learning Management Systems. In: *International Conference on Information Systems (ICIS)*, 2016, S. 1–21
- [Schreier 2010] SCHREIER, Margrit: Fallauswahl. In: MEY, Günter (Hrsg.); MRUCK, Katja (Hrsg.): *Handbuch qualitative Forschung in der Psychologie*. Wiesbaden: VS Verlag für Sozialwissenschaften, 2010, S. 238–251
- [Schröder 2002] SCHRÖDER, Hartwig: *Lernen - Lehren - Unterricht*. München [u.a.]: Oldenbourg Wissenschaftsverlag, 2002 (Hand- und Lehrbücher der Pädagogik)
- [Schryen et al. 2015] SCHRYEN, Guido; WAGNER, Gerit; BENLIAN, Alexander: Theory of Knowledge for Literature Reviews: An Epistemological Model, Taxonomy and Empirical Analysis of IS Literature. In: *International Conference on Information Systems (ICIS)*, 2015, S. 1–22
- [Schunk 1995] SCHUNK, Dale H.: Self-Efficacy and Education and Instruction. In: MADDUX, James E. (Hrsg.): *Self-Efficacy, Adaptation, and Adjustment*. Boston, MA: Springer US, 1995 (The Plenum Series in Social/Clinical Psychology), S. 281–303
- [Schwarzer u. Jerusalem 2002] SCHWARZER, Ralf; JERUSALEM, Matthias: Das Konzept der Selbstwirksamkeit. In: JERUSALEM, Matthias (Hrsg.); HOPF, Dieter (Hrsg.): *Selbstwirksamkeit und Motivationsprozesse in Bildungsinstitutionen*. Weinheim: Beltz, 2002 (Zeitschrift für Pädagogik, Beiheft, 44), S. 28–53

- [Seaborn u. Fels 2015] SEABORN, Katie; FELS, Deborah I.: Gamification in theory and action: A survey. In: *International Journal of Human-Computer Studies* 74 (2015), S. 14–31
- [Seaborn et al. 2013] SEABORN, Katie; PENNEFATHER, Peter; FELS, Deborah I.: Reimagining leaderboards: Towards gamifying competency models through social game mechanics. In: NACKE, Lennart E. (Hrsg.); HARRIGAN, Kevin (Hrsg.); RANDALL, Neil (Hrsg.): *Proceedings of the First International Conference on Gameful Design, Research, and Applications*, ACM, 2013 (Gamification '13), S. 107–110
- [Sedgewick u. Wayne 2014] SEDGEWICK, Robert; WAYNE, Kevin: *Algorithmen: Algorithmen und Datenstrukturen*. 4. aktualisierte Aufl. Hallbergmoos: Pearson, 2014 (Pearson Studium - IT)
- [Seel 1981] SEEL, Norbert M.: *Lernaufgaben und Lernprozesse*. Stuttgart [u.a.]: Kohlhammer, 1981
- [Seidewitz 2003] SEIDEWITZ, Ed: What models mean. In: *IEEE Software* 20 (2003), Nr. 5, S. 26–32
- [Sillaots 2014] SILLAOTS, Martin: Achieving Flow through Gamification: A study on Re-designing Research Methods Courses. In: BUSCH, Carsten (Hrsg.): *Proceedings of the 8th European Conference on Game Based Learning*, Academic Conferences and Publishing International Limited, 2014, S. 537–544
- [Simões et al. 2013] SIMÕES, Jorge; REDONDO, Rebeca D.; VILAS, Ana F.: A social gamification framework for a K-6 learning platform. In: *Computers in Human Behavior* 29 (2013), Nr. 2, S. 345–353
- [Simon 1996] SIMON, Herbert A.: *The sciences of the artificial*. 3rd ed. Cambridge, Mass.: MIT press, 1996
- [Singh 2016] SINGH, Ajit: *Filter Design Pattern*. <http://www.singhajit.com/filter-design-pattern/>. Version: 2016, Abruf: 28.05.2018
- [Smolka 2004] SMOLKA, Dieter: *Schülermotivation: Konzepte und Anregungen für die Praxis*. 2. durchges. Aufl. München: Luchterhand, 2004
- [Soley 2000] SOLEY, Richard; OBJECT MANAGEMENT GROUP (Hrsg.): *Model Driven Architecture*. Version: 2000. <http://www.omg.org/cgi-bin/doc?omg/00-11-05.pdf>, Abruf: 18.04.2018



- [Spinath 2011] SPINATH, Birgit: Lernmotivation. In: REINDERS, Heinz (Hrsg.); DITTON, Hartmut (Hrsg.); GRÄSEL, Cornelia (Hrsg.); GNIEWOSZ, Burkhard (Hrsg.): *Empirische Bildungsforschung*. Wiesbaden: VS Verlag für Sozialwissenschaften, 2011, S. 45–55
- [Spitzer 2003] SPITZER, Manfred: *Lernen: Gehirnforschung und die Schule des Lebens*. Heidelberg [u.a.]: Spektrum, Akademischer Verlag, 2003
- [Stahl u. Völter 2006] STAHL, Thomas; VÖLTER, Markus: *Model-driven software development: Technology, engineering, management*. Chichester [u.a.]: John Wiley, 2006
- [Stal u. Völter 2004] STAL, Michael; VÖLTER, Markus: *heise Developer - Episode 1: Über den Einsatz von Patterns*. <https://www.podcast.de/episode/293613822/Episode+1%3A+%C3%9Cber+den+Einsatz+von+Patterns/>. Version: 27.04.09, Abruf: 28.03.2018
- [Stampfl 2012] STAMPFL, Nora S.: *Die verspielte Gesellschaft: Gamification oder Leben im Zeitalter des Computerspiels*. Hannover: Heise, 2012
- [Stanford University 2016a] STANFORD UNIVERSITY: *Open Learning Initiative*. <http://oli.stanford.edu>. Version: 2016, Abruf: 25.05.2016
- [Stanford University 2016b] STANFORD UNIVERSITY: *Stanford Online Lagunita*. <https://lagunita.stanford.edu/>. Version: 2016, Abruf: 08.06.2016
- [Starke 2014] STARKE, Gernot: *Effektive Softwarearchitekturen: Ein praktischer Leitfaden*. 6. überarb. Aufl. München: Hanser, 2014
- [Steinberg et al. 2009] STEINBERG, Dave; BUDINSKY, Frank; PATERNOSTRO, Marcelo; MERKS, Ed: *EMF: Eclipse Modeling Framework*. 2nd ed., rev. and updated. Upper Saddle River, NJ: Addison-Wesley, 2009 (The eclipse series)
- [Stöcklin et al. 2014] STÖCKLIN, Nando; STEINBACH, Nico; SPANNAGEL, Christian: Computerunterstützte Gamification von Unterrichtseinheiten: Erste Erfahrungen mit QuesTanja. In: RENSING, Christoph (Hrsg.); TRAHASCH, Stephan (Hrsg.): *Proceedings of DeLFI Workshops 2014 co-located with 12th e-Learning Conference of the German Computer Society (DeLFI 2014)*, CEUR Workshop Proceedings, 2014, S. 270–277
- [Stöcklin et al. 2015] STÖCKLIN, Nando; STEINBACH, Nico; SPANNAGEL, Christian: QuesTanja: Kombination von individueller Förderung und sozialem Lernen in einem spielerischen Kontext. In: *Medienproduktion - Online-Zeitschrift für Wissenschaft und Praxis* (2015), Nr. 8, S. 2–4

- [Stott u. Neustaedter 2013] STOTT, Andrew; NEUSTAEDTER, Carman; CONNECTIONS LAB (Hrsg.): *Analysis of Gamification in Education*. Version: 2013. <http://clab.iat.sfu.ca/pubs/Stott-Gamification.pdf>, Abruf: 19.01.2016
- [Strembeck u. Zdun 2009] STREMBECK, Mark; ZDUN, Uwe: An approach for the systematic development of domain-specific languages. In: *Software: Practice and Experience* 39 (2009), Nr. 15, S. 1253–1292
- [Surendeleg et al. 2014] SURENDELEG, Garamkhand; MURWA, Violet; YUN, Han-Kyung; KIM, Yoon S.: The Role of Gamification in Education—A Literature Review. In: *Contemporary Engineering Sciences* 7 (2014), Nr. 29, 1609–1616. <http://dx.doi.org/10.12988/ces.2014.411217>, Abruf: 09.03.2015
- [Swacha 2014] SWACHA, Jakub: An Architecture of a Gamified Learning Management System. In: CAO, Yiwei (Hrsg.); VÄLJATAGA, Terje (Hrsg.); TANG, Jeff K. (Hrsg.); LEUNG, Howard (Hrsg.); LAANPERE, Mart (Hrsg.): *New Horizons in Web Based Learning*. Cham, Switzerland: Springer International Publishing, 2014 (Lecture Notes in Computer Science), S. 195–203
- [Sweet 1985] SWEET, Richard E.: The Mesa programming environment. In: PAYTON, Teri (Hrsg.); DEUTSCH, L. P. (Hrsg.); PURTILO, James (Hrsg.): *Proceedings of the ACM SIGPLAN 85 symposium on Language issues in programming environments*, ACM, 1985, S. 216–229
- [Takahashi 2008] TAKAHASHI, Dean: *Funware's threat to the traditional video game industry*. <http://venturebeat.com/2008/05/09/funwares-threat-to-the-traditional-video-game-industry/>. Version: 2008, Abruf: 06.05.2016
- [Tao et al. 2014] TAO, Ye; LIU, Guozhu; MOTTOK, Jürgen; HACKENBERG, Rudi; HAGEL, Georg: Just-in-Time Teaching in software engineering: A Chinese-German empirical case study. In: *Proceedings of IEEE Global Engineering Education Conference (EDUCON)*, IEEE, 2014, S. 983–986
- [Tao et al. 2015] TAO, Ye; LIU, Guozhu; MOTTOK, Jürgen; HACKENBERG, Rudi; HAGEL, Georg: Just-in-Time-Teaching experience in a Software Design Pattern course. In: *Proceedings of IEEE Global Engineering Education Conference (EDUCON)*, IEEE, 2015, S. 915–919
- [Thielscher 2010] THIELSCHER, Michael: A General Game Description Language for Incomplete Information Games. In: *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence and the Twenty-Second Innovative*

- Applications of Artificial Intelligence Conference*, AAAI Press, 2010, S. 994–999
- [Tolvanen 2005] TOLVANEN, Juha-Pekka: Domain-Specific Modeling for Full Code Generation. In: *Methods & Tools - Practical knowledge for the software developer, tester and project manager* 13 (2005), Nr. 3, 14–23. <http://www.methodsandtools.com/PDF/mt200503.pdf>, Abruf: 15.01.2015
- [Tolvanen u. Kelly 2010] TOLVANEN, Juha-Pekka; KELLY, Steven: Integrating models with domain-specific modeling languages. In: ROSSI, Matti (Hrsg.); TOLVANEN, Juha-Pekka (Hrsg.); SPRINKLE, Jonathan (Hrsg.); KELLY, Steven (Hrsg.): *Proceedings of the 10th Workshop on Domain-Specific Modeling*, ACM, 2010, S. 10:1–10:6
- [Tolvanen u. Kelly 2016] TOLVANEN, Juha-Pekka; KELLY, Steven: Model-Driven Development Challenges and Solutions - Experiences with Domain-Specific Modeling in Industry. In: *Proceedings of the 4th International Conference on Model-Driven Engineering and Software Development*, SCITEPRESS - Science and Technology Publications, 2016, S. 711–719
- [Tsichritzis 1997] TSICHRITZIS, Dennis: The Dynamics of Innovation. In: DENNING, Peter J. (Hrsg.); METCALFE, Robert M. (Hrsg.): *Beyond Calculation*. New York, NY: Springer New York, 1997, S. 259–265
- [Udacity Inc. 2016] UDACITY INC.: *Udacity*. <https://www.udacity.com/>. Version: 2016, Abruf: 25.05.2016
- [Udemy Inc. 2016] UDEMY INC.: *Udemy*. <https://www.udemy.com/>. Version: 2016, Abruf: 25.05.2016
- [Ulich 1996] ULICH, Klaus: *Beruf: Lehrer-in: Arbeitsbelastungen, Beziehungskonflikte, Zufriedenheit*. Weinheim [u.a.]: Beltz, 1996 (Beltz grüne Reihe)
- [Ullenboom 2017] ULLENBOOM, Christian: *Java ist auch eine Insel*. 13. Aufl. Bonn: Rheinwerk Verlag, 2017
- [van Broeckhoven u. de Troyer 2013] VAN BROECKHOVEN, Frederik; DE TROYER, Olga: ATTAC-L: A modeling language for educational virtual scenarios in the context of preventing cyber bullying. In: *2013 IEEE 2nd International Conference on Serious Games and Applications for Health (SeGAH)*, IEEE, 2013, S. 1–8
- [van Lamsweerde 2013] VAN LAMSWEERDE, Axel: *Requirements engineering: From system goals to UML models to software specifications*. Chichester: Wiley, 2013

- [van Roy u. Zaman 2017] VAN ROY, Rob; ZAMAN, Bieke: Why Gamification Fails in Education and How to Make It Successful: Introducing Nine Gamification Heuristics Based on Self-Determination Theory. In: MA, Minhua (Hrsg.); OIKONOMOU, Andreas (Hrsg.): *Serious Games and Edutainment Applications*. Cham: Springer, 2017, S. 485–509
- [Verkoeijen et al. 2006] VERKOEIJEN, Peter P. J. L.; RIKERS, Remy M. J. P.; TE WINKEL, Wilco W. R.; VAN DEN HURK, Marianne M.: Do student-defined learning issues increase quality and quantity of individual study? In: *Advances in health sciences education: theory and practice* 11 (2006), Nr. 4, S. 337–347
- [Viyović et al. 2014] VIYOVIĆ, Vladimir; MAKSIMOVIĆ, Mirjana; PERISIĆ, Branko: Sirius: A rapid development of DSM graphical editor. In: *18th International Conference on Intelligent Engineering Systems INES*, IEEE, 2014, S. 233–238
- [Völter et al. 2013] VÖLTER, Markus; BENZ, Sebastian; DIETRICH, Christian; ENGELMANN, Birgit; HELANDER, Mats; KATS, Lennart; VISSER, Eelco; WACHSMUTH, Guido: *DSL engineering: Designing, implementing and using domain-specific languages*. North Charleston: CreateSpace Independent Publishing Platform, 2013
- [Waldherr u. Walter 2009] WALDHERR, Franz; WALTER, Claudia: *Didaktisch und Praktisch: Ideen und Methoden für die Hochschullehre*. Stuttgart: Schäffer-Poeschel Verlag, 2009
- [Watkins u. Mazur 2010] WATKINS, Jessica; MAZUR, Eric: Just-in-Time Teaching and Peer Instruction. In: SIMKINS, Scott (Hrsg.); MAIER, Mark H. (Hrsg.): *Just-in-Time Teaching*. Sterling, Virginia: Stylus Publishing, 2010, S. 39–62
- [Weinert 1999] WEINERT, Franz E.; MAX PLANCK INSTITUT FÜR PSYCHOLOGISCHE FORSCHUNG (Hrsg.): *Concepts of Competence: Definition and Selection of Competencies*. München, 1999
- [Weinert 2001] WEINERT, Franz E.: *Leistungsmessungen in Schulen*. 2. Aufl. Weinheim [u.a.]: Beltz-Verlag, 2001 (Beltz Pädagogik)
- [Weinstein u. Mayer 1986] WEINSTEIN, Claire E.; MAYER, Richard E.: The Teaching of Learning Strategies. In: WITTRICK, Merlin C. (Hrsg.): *Handbook of research on teaching*. New York [u.a.]: Collier Macmillan Publishers, 1986, S. 315–327

- [Weiss u. Lai 1999] WEISS, David M.; LAI, Chi Tau R.: *Software product-line engineering: A family-based software development process*. Reading, Mass.: Addison-Wesley, 1999
- [Werbach 2014] WERBACH, Kevin: (Re) Defining Gamification: A Process Approach. In: SPAGNOLLI, Anna (Hrsg.); CHITTARO, Luca (Hrsg.); GAMBERINI, Luciano (Hrsg.): *Persuasive Technology*. Cham: Springer, 2014, S. 266–272
- [Werbach u. Hunter 2012] WERBACH, Kevin; HUNTER, Dan: *For the win: How game thinking can revolutionize your business*. Philadelphia: Wharton Digital Press, 2012
- [Werbach u. Hunter 2015] WERBACH, Kevin; HUNTER, Dan: *The Gamification Toolkit: Dynamics, Mechanics, and Components for the Win*. New York: Wharton Digital Press, 2015
- [Wittgenstein 2001] WITTGENSTEIN, Ludwig: *Logisch-philosophische Abhandlung: Kritische Edition = Tractatus logico-philosophicus*. Bd. 1359. 2. Aufl. Frankfurt am Main: Suhrkamp, 2001
- [Woolfolk 2014] WOOLFOLK, Anita: *Pädagogische Psychologie*. 12. aktual. Aufl. Hallbergmoos: Pearson Studium, 2014
- [Wyner et al. 2010] WYNER, Adam; ANGELOV, Krasimir; BARZDINS, Guntis; DAMLJANOVIC, Danica; DAVIS, Brian; FUCHS, Norbert; HOEFLER, Stefan; JONES, Ken; KALJURAND, Kaarel; KUHN, Tobias; LUTS, Martin; POOL, Jonathan; ROSNER, Mike; SCHWITTER, Rolf; SOWA, John: On Controlled Natural Languages: Properties and Prospects. In: HUTCHISON, David (Hrsg.); KANADE, Takeo (Hrsg.); KITTLER, Josef (Hrsg.); KLEINBERG, Jon M. (Hrsg.); MATTERN, Friedemann (Hrsg.); MITCHELL, John C. (Hrsg.); NAOR, Moni (Hrsg.); NIERSTRASZ, Oscar (Hrsg.); PANDU RANGAN, C. (Hrsg.); STEFFEN, Bernhard (Hrsg.); SUDAN, Madhu (Hrsg.); TERZOPOULOS, Demetri (Hrsg.); TYGAR, Doug (Hrsg.); VARDI, Moshe Y. (Hrsg.); WEIKUM, Gerhard (Hrsg.); FUCHS, Norbert E. (Hrsg.): *Controlled Natural Language* Bd. 5972. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, S. 281–289
- [Yee 2005] YEE, Nick: Motivations of Play in MMORPGs. In: CASTELL, Suzanne de (Hrsg.); JENSON, Jennifer (Hrsg.): *Proceedings of Digital Games Research Association (DiGRA): Changing Views - Worlds in Play*, 2005
- [Yohannis 2016] YOHANNIS, Alfa: Gamification of Software Modelling Learning. In: NEJATI, Shiva (Hrsg.); SALAY, Rick (Hrsg.): *Proceedings of the Doctoral Symposium at the 19th ACM/IEEE International Conference of Model-Driven Engineering Languages and Systems 2016*, 2016, S. 1–7

- [Zagal et al. 2005] ZAGAL, José P.; MATEAS, Michael; FERNÁNDEZ-VARA, Clara; HOCHHALTER, Brian; LICHTI, Nolan: Towards an Ontological Language for Game Analysis. In: CASTELL, Suzanne de (Hrsg.); JENSON, Jennifer (Hrsg.): *Proceedings of Digital Games Research Association (DiGRA): Changing Views - Worlds in Play*, 2005, S. 3–14
- [Zelewski 2007] ZELEWSKI, Stephan: Kann Wissenschaftstheorie behilflich für die Publikationspraxis sein? In: LEHNER, Franz (Hrsg.); ZELEWSKI, Stephan (Hrsg.): *Wissenschaftstheoretische Fundierung und wissenschaftliche Orientierung der Wirtschaftsinformatik*. Berlin: Gito, 2007, S. 71–120
- [Zichermann u. Cunningham 2011] ZICHERMANN, Gabe; CUNNINGHAM, Christopher: *Gamification by Design - Implementing Game Mechanics in Web and Mobile Apps*. Sebastopol, CA: O'Reilly Media, Inc, 2011
- [Zichermann u. Linder 2013] ZICHERMANN, Gabe; LINDER, J.: *The Gamification Revolution: How Leaders Leverage Game Mechanics to Crush the Competition*. New York [u.a.]: McGraw-Hill Education, 2013
- [Zimmerman 2000] ZIMMERMAN, Barry J.: Self-Efficacy: An Essential Motive to Learn. In: *Contemporary educational psychology* 25 (2000), Nr. 1, S. 82–91



---

## ANHÄNGE

---

### Inhaltsübersicht:

- Anhang A.1: Der Gamified Course Design Record
- Anhang A.2: Schema zur Lernplattformanalyse
- Anhang A.3: Fragebogen zur Findung von Notationsvorschlägen für die Emendo DSML
- Anhang A.4: Verbindungsübersicht über Notationselemente
- Anhang A.5: Anforderungen an das Emendo LMS
- Anhang A.6: Screenshots Hauptseiten des Emendo LMS
- Anhang A.7: Screenshots des Emendo LMS Mobile
- Anhang A.8: Geheft für die zweite Evaluationsstufe
- Anhang A.9: Fragenbogen für Evaluationsteilnehmende

## A.1 DER GAMIFIED COURSE DESIGN RECORD

Gamified Course Design Record		
Allgemeine Informationen		
Name LLA: (Name des LLA)		
Datum: (Datum)	Lehrveranstaltung: (Name LV)	
Kurzbeschreibung: (Gesamtübersichtsbeschreibung)		
Vorkenntnisse Lernende (notwendig und optional):		
Lernziele (fachlich)*:		Kompetenz(en):
fLZ1	(Template-basiertes Lernziel)	(Name Kompetenz)
fLZ2		
fLZ3		
fLZ4		
Lernziele (überfachlich):		
üFL1	(Template-basiertes Lernziel)	
üLZ2		
üLZ3		
üLZ4		
Methoden:	Integration in LLA	Für Lernziel(e)
ME1: (Name Methode)	(Integrationsbeschreibung)	(Lernzielkürzel)
ME2:		
ME3:		

Game Design Elemente			
Dynamics:			
1	Narrative:		
	(Kurzbeschreibung)		
	Beziehung zu anderen GDEs:		Lernstufe**:
			Zitiert nach Roth, H. (1983). Pädagogische Psychologie des Lehrens und Lernens (16th ed.). Hannover: Schroedel Schulbuchverlag. S. 208ff.
	M1: (Name + Operationalisierungsbeschreibung und notwendige Aktionen)	(GDE + Art der Beziehung)	1. Lernschritt: „Stufe der Motivation ...
	C1:		
	M2:		2. Lernschritt: „Stufe der Schwierigkeiten
	C2:		
	M3:		3. Lernschritt: „Stufe der Lösung...
Mechanic/Component	C3:		
	M4:		4. Lernschritt: „Stufe des Tuns und Ausführens...
	C4:		
	M5:		5. Lernschritt: „Stufe des Behaltens und Einübens...



		C5:			
		M6:			6. Lernschritt: „Stufe des Bereitstellens, der Übertragung und der Integration des Gelernten ...“
		C6:			
2	Emotions:				
	(Kurzbeschreibung)				
		Beziehung zu anderen GDEs:	Motivationspsychologische Theorie (Bestandteil)***:		
	M1: (Name + Operationalisierungsbeschreibung)	(GDE + Art der Beziehung)	(Nennung der Theorie oder des Theoriebestandteils, z.B. SDT: Autonomie)		
	C1:				
	M2:				
	C2:				
	M3:				
	C3:				
	M4:				
	C4:				
3	Relationships:				
	(Kurzbeschreibung)				
		Beziehung zu anderen GDEs:			
	M1: (Name + Operationalisierungsbeschreibung)	(GDE + Art der Beziehung)			
	C1:				
	M2:				
	C2:				
	M3:				
	C3:				
	M4:				
	C4:				
4	Progression:				
	(Kurzbeschreibung)				
		Beziehung zu anderen GDEs:	Art der Fortschrittsmetrik:		
	M1: (Name + Operationalisierungsbeschreibung + ggf. Mapping auf Kompetenzniveaus)	(GDE + Art der Beziehung)	(quantitativ / qualitativ)		
	C1:				
	M2:				
	C2:				
	M3:				
	C3:				
	M4:				
	C4:				



## A.2 SCHEMA ZUR LERNPLATTFORMANALYSE

## Schema zur Charakterisierung von Game-Design-Elementen (GDE)

<b>Bezeichnung:</b>			
<b>Plattform URL:</b>			
<b>Eigenschaften / Attribute:</b>	<b>Bezeichnung</b>	<b>Ausprägungen</b>	
<b>Funktionen:</b>	<b>Bezeichnung</b>		
<b>Beziehungen zu anderen Game-Design-Elementen:</b>	<b>Bezeichnung des Game-Design-Elements</b>		<b>Art der Beziehung</b>
<b>Verwendungskontext:</b>	<b>Beschreibung</b>	<b>Taxonomiestufe</b>	<b>Typische Aufgabentypen</b>

### A.3 FRAGEBOGEN ZUR FINDUNG VON NOTATIONSVORSCHLÄGEN FÜR DIE EMENDO DSML

## Dozierendenbefragung – Notationsfindung für eine domänenspezifische Modellierungssprache

Bitte versetzen Sie sich in den Kontext eines Benutzers eines grafischen Modellierungswerkzeugs, durch welches eine **domänenspezifische Modellierungssprache** (DSML) für den **Lehr-Lernkontext** bereitgestellt wird. Mit der DSML ist es möglich unterschiedliche Typen von Lernaufgaben für Studierende zu modellieren und dabei Aufgaben in einer bestimmten Abfolge anzuordnen oder sequenzlos. In letzterer Anordnung obliegt die Entscheidung dem Lernenden, in welcher Reihenfolge die Lernaufgaben bearbeitet werden. Die Bearbeitung von Lernaufgaben ist mit spielifizierten Elementen (Game Design Elementen) über Regeln verbunden. Dies bedeutet, dass Aktionen, die von Lernenden mit Lernaufgaben ausgeführt werden (z.B. eine Aufgabe beantworten) Konsequenzen besitzen – entweder Game Design Elemente (GDE) wie Punkte oder Badges, womit Lernende „belohnt“ werden oder textuelle Feedbacks, die die Konsequenz einer Aktion Lernenden direkt mitteilen. Das Verhalten, also welche Bedingungen welche Konsequenzen zur Folge haben, wird über Regeln definiert.

Als zu Grunde liegende Methode eines Lehr-Lernarrangements wird **JITT** („Just-in-Time Teaching“) gewählt.

Die folgende Tabelle zeigt für die DSML relevante Domänenkonzepte. Bitte **skizzieren** Sie **freihändig** in der rechten Spalte, wie Ihrer Meinung nach eine intuitive grafische Notation (z.B. Symbol) zum jeweiligen Domänenkonzept aussehen könnte. Wenn Sie sich in einem Feld unsicher sind, lassen sie das jeweilige Feld bitte leer.

Sinn und Zweck dieser Dozierendenbefragung besteht darin, dass „**not-invented-here**“-**Phänomen zu vermeiden**, welches auftreten kann, wenn potentielle Benutzer einer Software nicht in deren Erstellprozess involviert sind, um beispielsweise Ihre Vorschläge und Ideen einbringen zu können. Ebenso soll für den Ersteller der Software eine Vorstellung darüber gewonnen werden, wie mit der Domäne vertraute Benutzer, unterschiedliche Konzepte, grafisch repräsentieren würden. Die Befragungen werden durch den Ersteller der Software ausgewertet. Dabei wird angestrebt über alle Befragungen hinweg den „größten gemeinsamen Nenner“ einer grafischen Notation für ein Domänenkonzept zu identifizieren.

Domänenkonzept	Grafische Notation
Pflichtverbindung zwischen zwei Modellelementen (z.B. kausale Abhängigkeit zwischen zwei Aufgaben A und B, für die gilt: wenn A erfüllt, dann erst B erfüllbar)	

Optionalverbindung zwischen zwei Modellelementen (keine kausale Abhängigkeit, damit keine zwingende Bearbeitungsreihenfolge)	
Zuordnungsverbindung zwischen zwei Modellelementen (z.B. Antwortmöglichkeit einer Single Choice Aufgabe zu einer Single Choice Aufgabe)	
Gruppierung (zum Gruppieren von Aufgaben, um beispielsweise für eine Gruppe von Aufgaben eine Regel zu definieren, welche für alle Elemente der Gruppe gilt)	
Game Design Element Punktesystem	

Game Design Element Badge	
Game Design Element Skill	
Game Design Element Levelsystem	
Game Design Element (Content-) Unlock (Freischaltung von Lernelement)	

Textuelles Feedback als Reaktion auf eine Benutzeraktion	
Belohnungsregel (bestehend aus mind. einer Bedingung und mind. einem Game Design Element)	
Feedbackregel (bestehend aus mind. einer Bedingung und mind. einem Feedback)	
Lernmaterial: Textblock (kann Grafiken und Links enthalten)	

Lernmaterial: Video	
Lernmaterial: Podcast	
Lösungshinweis zu einer Aufgabe	
Aufgabe: Freitextaufgabe (Benutzer antwortet in Form von Freitext auf eine Aufgabenstellung)	



Aufgabe: Single Choice Aufgabe	
Aufgabe: Multiple Choice Aufgabe	
Aufgabe: Eingabeaufgabe (Benutzer muss z.B. eine Einschätzung in Form einer Ganzzahl abgeben)	
Antwortmöglichkeit: Single Choice Antwort	

Antwortmöglichkeit: Multiple Choice Antwort	
---	--

Vielen Dank für Ihre Teilnahme. Bei Fragen stehe ich Ihnen gerne jederzeit per E-Mail zur Verfügung.  
Sie erreichen mich unter der folgenden E-Mail Adresse: [alexander.bartel@hs-kempten.de](mailto:alexander.bartel@hs-kempten.de)

Es grüßt Sie herzlich

Alexander Bartel

## A.4 VERBINDÜNGSÜBERSICHT ÜBER NOTATIONSELEMENTE

Lernelement		Lernelement												Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement			Lernelement</		
-------------	--	-------------	--	--	--	--	--	--	--	--	--	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	-------------	--	--	---------------	--	--

## A.5 ANFORDERUNGEN AN DAS EMENDO LMS

**Funktionale Anforderungen:**

<b>Nr.</b>	<b>Funktionale Anforderung</b>	<b>Klassifikation</b>
1	Das Emendo LMS wird Lehrenden und Lernenden die Möglichkeit bieten, sich anzumelden.	Basisfaktor
2	Das Emendo LMS wird Lehrenden und Lernenden die Möglichkeit bieten, sich abzumelden.	Basisfaktor
3	Das Emendo LMS wird Lehrenden und Lernenden die Möglichkeit bieten, deren Anmeldedaten zu ändern.	Leistungsfaktor
4	Das Emendo LMS muss Lehrenden und Lernenden die Möglichkeit bieten, deren Profil anzuzeigen.	Basisfaktor
5	Das Emendo LMS wird Lehrenden und Lernenden die Möglichkeit bieten, deren Profilbild zu ändern.	Begeisterungsfaktor
6	Das Emendo LMS muss Lehrenden und Lernenden die Möglichkeit bieten, die Übersicht der Diskussionsbeiträge anzuzeigen.	Basisfaktor
7	Das Emendo LMS muss Lernenden die Möglichkeit bieten, Diskussionsbeiträge zu Lernbausteinen zu erstellen.	Leistungsfaktor
8	Das Emendo LMS muss Lehrenden und Lernenden die Möglichkeit bieten, Diskussionsbeiträge zu löschen, welche sie selbst erstellt haben.	Leistungsfaktor
9	Das Emendo LMS wird Lehrenden und Lernenden die Möglichkeit bieten, Diskussionsbeiträge als hilfreich zu markieren.	Leistungsfaktor
10	Das Emendo LMS sollte Lehrenden und Lernenden die Möglichkeit bieten, Diskussionsbeiträge zu Lernbausteinen zu kommentieren.	Leistungsfaktor
weiter auf nächster Seite		

<b>Nr.</b>	<b>Funktionale Anforderung</b>	<b>Klassifikation</b>
11	Das Emendo LMS sollte Lehrenden und Lernenden die Möglichkeit bieten, Diskussionskommentare zu löschen, welche sie selbst erstellt haben.	Leistungsfaktor
12	Das Emendo LMS muss Lehrenden und Lernenden die Möglichkeit bieten, Diskussionskommentare als hilfreich zu markieren.	Leistungsfaktor
13	Das Emendo LMS muss dem Lernenden die Möglichkeit bieten, die Bestenliste anzuzeigen.	Leistungsfaktor
14	Das Emendo LMS muss dem Lernenden die Möglichkeit bieten, die Bestenliste nach Punktestand oder Level zu sortieren.	Leistungsfaktor
15	Das Emendo LMS muss dem Lernenden die Möglichkeit bieten, die Übersicht der Lernbausteine anzuzeigen.	Basisfaktor
16	Das Emendo LMS muss in der Übersicht der Lernbausteine darstellen, welche Lernbausteine zur Bearbeitung freigeschaltet bzw. nicht freigeschaltet sind.	Begeisterungsfaktor
17	Das Emendo LMS muss dem Lernenden die Möglichkeit bieten, anzuzeigen, welche Lernbausteine beim erfolgreichen Abschluss eines Lernbausteins freigeschaltet werden.	Begeisterungsfaktor
18	Das Emendo LMS muss zu jedem Lernbaustein Detailinformationen anzeigen. (Typ, Bearbeitungsdauer, Belohnungen, usw.)	Begeisterungsfaktor
19	Das Emendo LMS muss dem Lernenden die Möglichkeit bieten, zum nächsten freigeschalteten Lernbaustein zu navigieren.	Begeisterungsfaktor
20	Das Emendo LMS muss dem Lernenden die Möglichkeit bieten, Aufgaben zu bearbeiten.	Basisfaktor
21	Das Emendo LMS muss dem Lernenden die Möglichkeit bieten, Lösungen zu Aufgaben abzugeben.	Basisfaktor

weiter auf nächster Seite

<b>Nr.</b>	<b>Funktionale Anforderung</b>	<b>Klassifikation</b>
22	Das Emendo LMS muss alle Lösungsabgaben zu Aufgaben (außer Freitextabgaben) automatisch auf Richtigkeit überprüfen.	Leistungsfaktor
23	Das Emendo LMS muss nach dem Abschluss eines Lernbausteins die angehängten Regeln überprüfen.	Begeisterungsfaktor
24	Das Emendo LMS muss die Vergabe von Belohnungen oder den Erhalt von Feedbacks dem Lernenden anzeigen.	Leistungsfaktor
25	Das Emendo LMS muss dem Lernenden die Möglichkeit bieten, Textblöcke zu lesen.	Basisfaktor
26	Das Emendo LMS muss dem Lernenden die Möglichkeit bieten, Textblöcke innerhalb von Aufgaben zu lesen.	Basisfaktor
27	Das Emendo LMS muss Quellcode in einem Textblock formatiert anzeigen.	Begeisterungsfaktor
28	Das Emendo LMS muss dem Lernenden die Möglichkeit bieten, Podcasts abzuspielen.	Basisfaktor
29	Das Emendo LMS muss dem Lernenden die Möglichkeit bieten, Podcasts innerhalb von Aufgaben abzuspielen.	Basisfaktor
30	Das Emendo LMS muss dem Lernenden die Möglichkeit bieten, Videoclips abzuspielen.	Basisfaktor
31	Das Emendo LMS muss dem Lernenden die Möglichkeit bieten, Videoclips innerhalb von Aufgaben abzuspielen.	Basisfaktor
32	Das Emendo LMS muss den Bearbeitungszustand von Lernbausteinen der Lernenden automatisch speichern.	Basisfaktor
33	Das Emendo LMS muss den Bearbeitungszustand von Lernbausteinen der Lernenden automatisch laden.	Basisfaktor
weiter auf nächster Seite		

<b>Nr.</b>	<b>Funktionale Anforderung</b>	<b>Klassifikation</b>
34	Das Emendo LMS muss dem Lehrenden die Möglichkeit bieten, den aktuellen Bearbeitungsfortschritt aller Lernbausteine der Lernenden anzuzeigen.	Leistungsfaktor
35	Das Emendo LMS muss dem Lehrenden die Möglichkeit bieten, die Freitextabgaben der Lernenden anzuzeigen.	Basisfaktor
36	Das Emendo LMS muss dem Lehrenden die Möglichkeit bieten, Freitextabgaben der Lernenden anhand von Kriterien zu bewerten.	Basisfaktor
37	Das Emendo LMS muss dem Lehrenden die Möglichkeit bieten, die Freitextabgaben der Lernenden als Powerpoint-Datei zu exportieren.	Begeisterungsfaktor

#### **Nichtfunktionale Anforderungen:**

<b>Nr.</b>	<b>Nichtfunktionale Anforderung</b>	<b>Typ</b>
1	Der Umgang mit dem Emendo LMS muss für Lehrende und Lernende leicht zu erlernen sein.	Erlernbarkeit (ISO/IEC 9126)
2	Jeder im Emendo LMS freigeschaltete Lernbaustein muss innerhalb von maximal 3 Klicks (ausgenommen Auswahl von Mehrfachantworten) vollständig bearbeitbar sein, ausgehend von der Übersicht aller Lernbausteine.	Bedienbarkeit (ISO/IEC 9126)
3	Alle Reaktionen auf Benutzereingaben dürfen nicht länger als 500ms in Anspruch nehmen.	Bedienbarkeit (ISO/IEC 9126)
4	Das Emendo LMS muss eine zeitgemäße und für Benutzer ansprechende Benutzeroberfläche besitzen (z.B. unter Benutzung des GUI-Frameworks WebLaF).	Attraktivität (ISO/IEC 9126)

weiter auf nächster Seite

<b>Nr.</b>	<b>Nichtfunktionale Anforderung</b>	<b>Typ</b>
5	Das Emendo LMS muss Aufgabenabgaben außer Freitextabgaben innerhalb von max. 500 ms auf deren Korrektheit überprüfen und das Ergebnis der Überprüfung anzeigen.	Zeitverhalten (ISO/IEC 9126)
6	Das Emendo LMS muss das Sortieren der Bestenliste nach Punktestand oder Level in weniger als 500ms vornehmen, bei maximal 100 Einträgen.	Zeitverhalten (ISO/IEC 9126)
7	Das Emendo LMS muss gewährleisten, dass alle Lernfortschritte des Lernenden (auch nach einem Absturz der Software) persistent und konsistent gespeichert sind.	Wiederherstellbarkeit (ISO/IEC 9126)
8	Das Emendo LMS muss eine Adaption auf weitere Java-basierte Technologieplattformen unterstützen. Im Speziellen sind zu nennen: Android ab Version 4.0-aktuell; Java EE 6-aktuell.	Anpassbarkeit (ISO/IEC 9126)
9	Das Emendo LMS muss eine leichte Erweiterung der Systemfunktionalitäten, vorallem in Bezug auf Views, gewährleisten.	Modifizierbarkeit (ISO/IEC 9126)
10	Das Emendo LMS muss logging-Mechanismen einsetzen, um auftretende Fehler zu dokumentieren und nachvollziehbar zu machen (z.B. log4j).	Analysierbarkeit (ISO/IEC 9126)
11	Das Emendo LMS muss alle dynamischen Daten (Zustände von Lernelementen) des Lernenden in einer SQLite Datenbank in der Version 3.16.1 verwalten.	Technologische Randbedingung
12	Das Emendo LMS muss in Java Version 7 programmiert sein.	Technologische Randbedingung
13	Das Emendo LMS muss auf einer x64 Windows 7 (oder höher) Umgebung lauffähig sein.	Technologische Randbedingung

weiter auf nächster Seite



<b>Nr.</b>	<b>Nichtfunktionale Anforderung</b>	<b>Typ</b>
14	Das Emendo LMS muss objektrelationales Mapping (z.B. mit ORMLite) für den Zugriff und die Manipulation der Daten in der Datenbank unterstützen.	Technologische Randbedingung
15	Das Emendo LMS muss für die Formatierung von Quellcode innerhalb von Textblöcken die Library RsyntaxTextarea verwenden.	Technologische Randbedingung
16	Die Benutzeroberfläche des Emendo LMS muss in deutscher Sprache umgesetzt sein.	Kulturelle Randbedingung

### Glossar:

<b>Begriff</b>	<b>Bedeutung</b>
Aufgabe	Unter einer Aufgabe versteht man Singlechoice, Multiplechoice, Texteingabe und Freitext Fragestellungen.
Videoclip	Unter einem Videoclip versteht man eine Bewegtbildaufnahme mit optionalem Tonmaterial. Diese kann mit UI-Steuerungselementen abgespielt, pausiert, vor- und zurückgespult werden.
Podcast	Unter einem Podcast versteht man eine Tonaufnahme, gängigerweise die Stimme eines Sprechers oder mehrerer Sprecher. Diese kann mit UI-Steuerungselementen abgespielt, pausiert, vor- und zurückgespult werden.
Textblock	Unter einem Textblock versteht man die Aneinanderreihung mehrerer Sätze. Beim Auftreten von Programmcode-Elementen innerhalb des Textblocks wird dieser, abhängig von der dargestellten Programmiersprache, entsprechend hervorgehoben.
Lernmaterial	Unter einem Lernmaterial versteht man Videoclips, Podcasts und Textblöcke. Auf Lernmaterialien kann im Gegensatz zu Aufgaben keine Antwort abgegeben werden.
weiter auf nächster Seite	

<b>Begriff</b>	<b>Bedeutung</b>
Lernbaustein	Unter einem Lernbaustein versteht man sowohl Aufgaben als auch Lernmaterialien.
Anmeldedaten	Unter Anmeldedaten versteht man die Kombination aus Benutzername und Passwort.
Lernende	Unter Lernenden versteht man die Benutzergruppe, die das Programm zum Lernen nutzen. Lernende können beispielsweise Studenten sein.
Lehrende	Unter Lehrenden versteht man die Benutzergruppe, die das Programm zum Lehren nutzen. Zu der Benutzergruppe gehören zum Beispiel Dozenten, Professoren und Übungsgruppenleiter.

## A.6 SCREENSHOTS HAUPTSEITEN DES EMENDO LMS

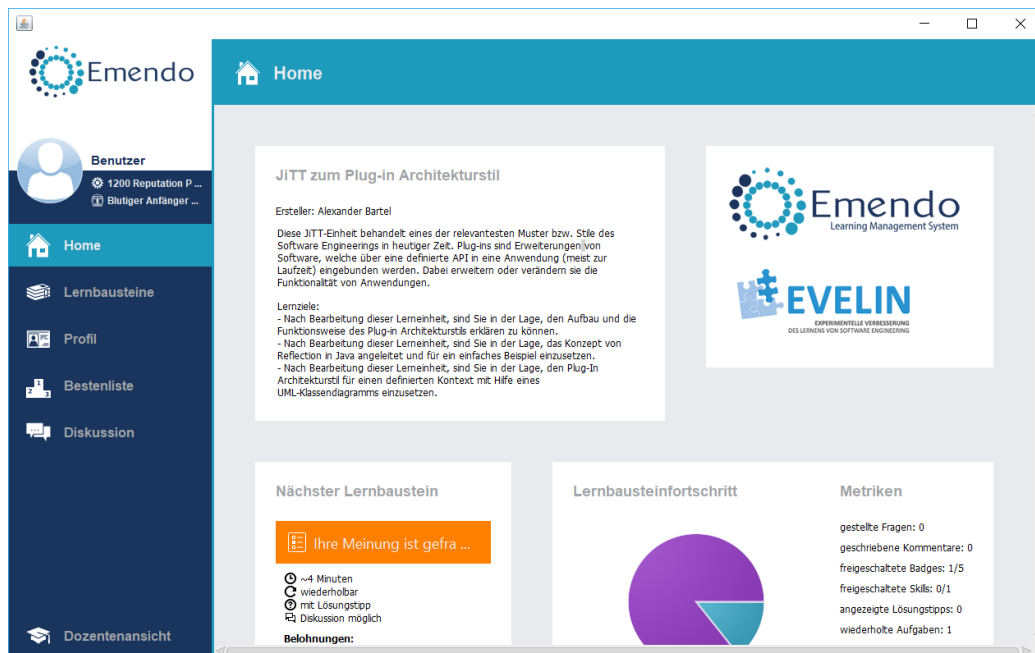


Abbildung 96: Screenshot Bildschirmseite Home

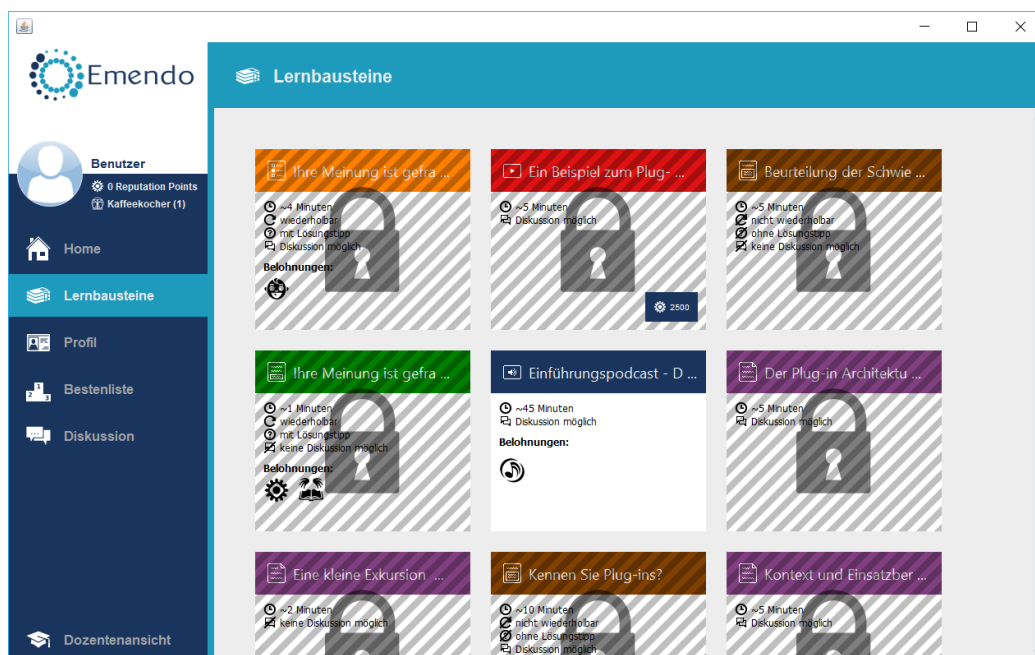


Abbildung 97: Screenshot Bildschirmseite Lernbausteine

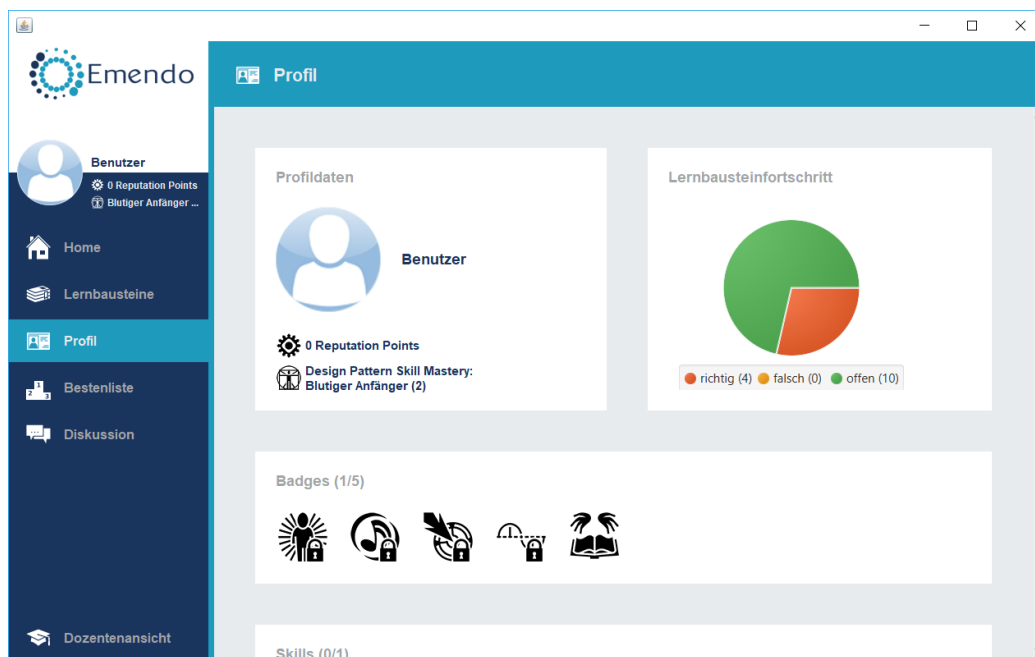


Abbildung 98: Screenshot Bildschirmseite Profil

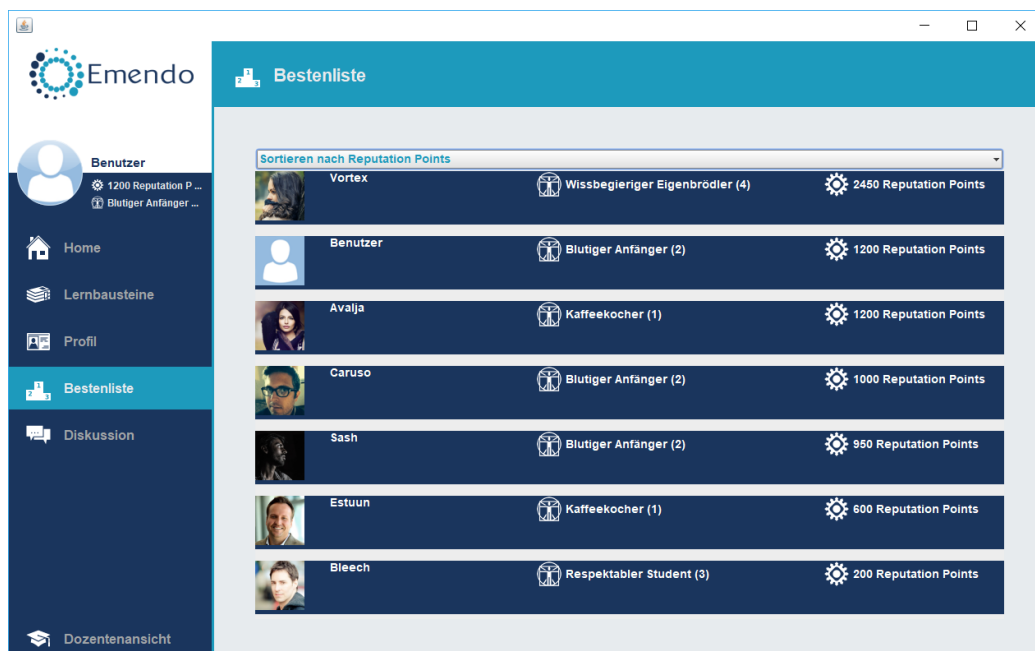


Abbildung 99: Screenshot Bildschirmseite Bestenliste

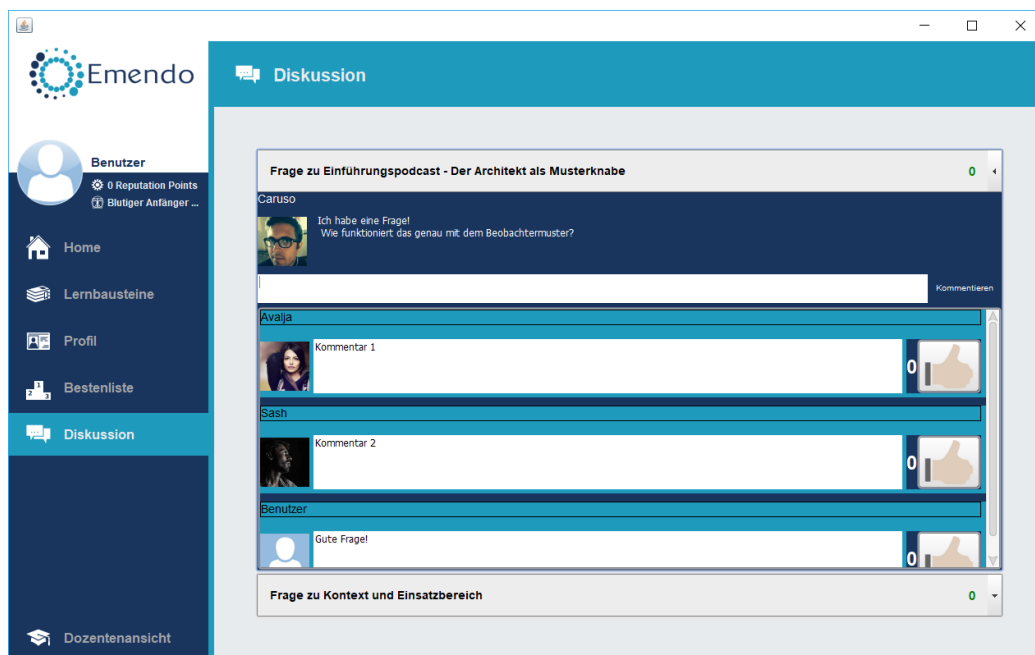


Abbildung 100: Screenshot Bildschirmseite Diskussion

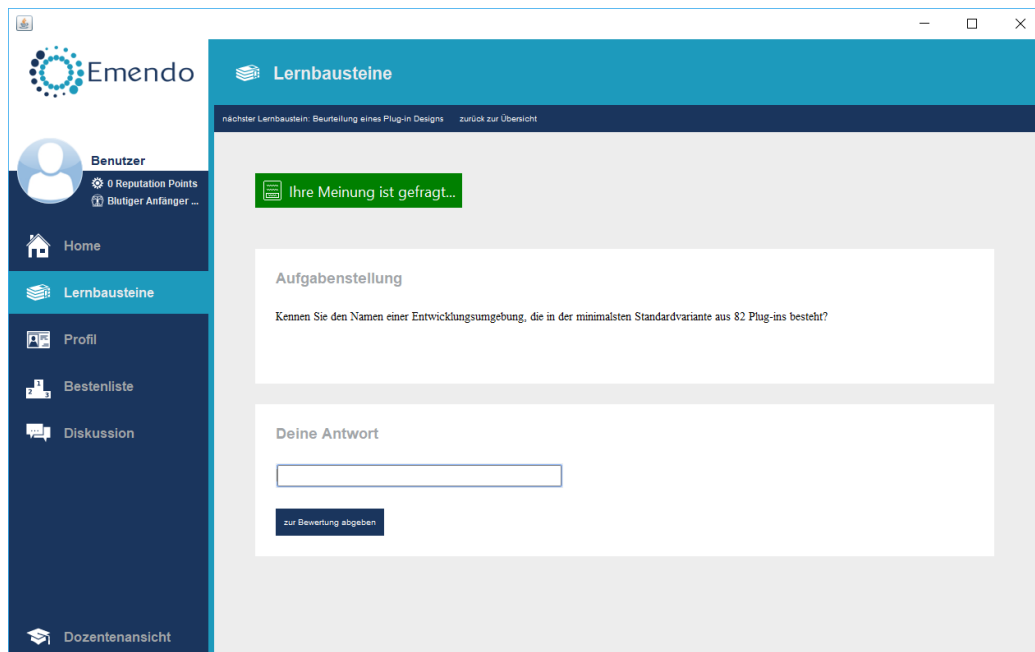


Abbildung 101: Screenshot Bearbeitung Eingabe-Aufgabe

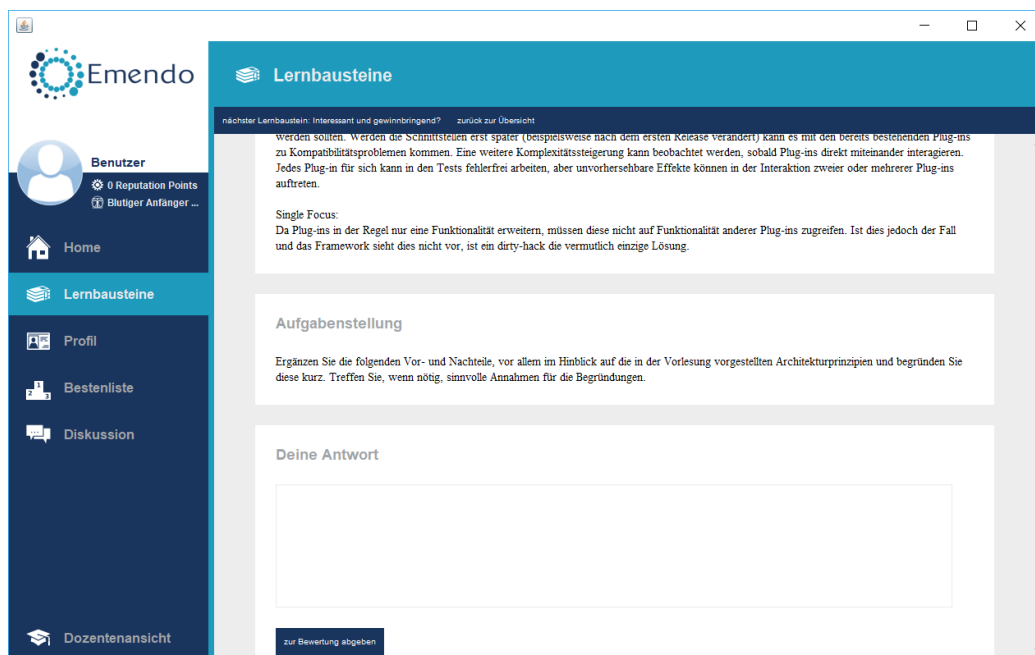


Abbildung 102: Screenshot Bearbeitung Freitext-Aufgabe

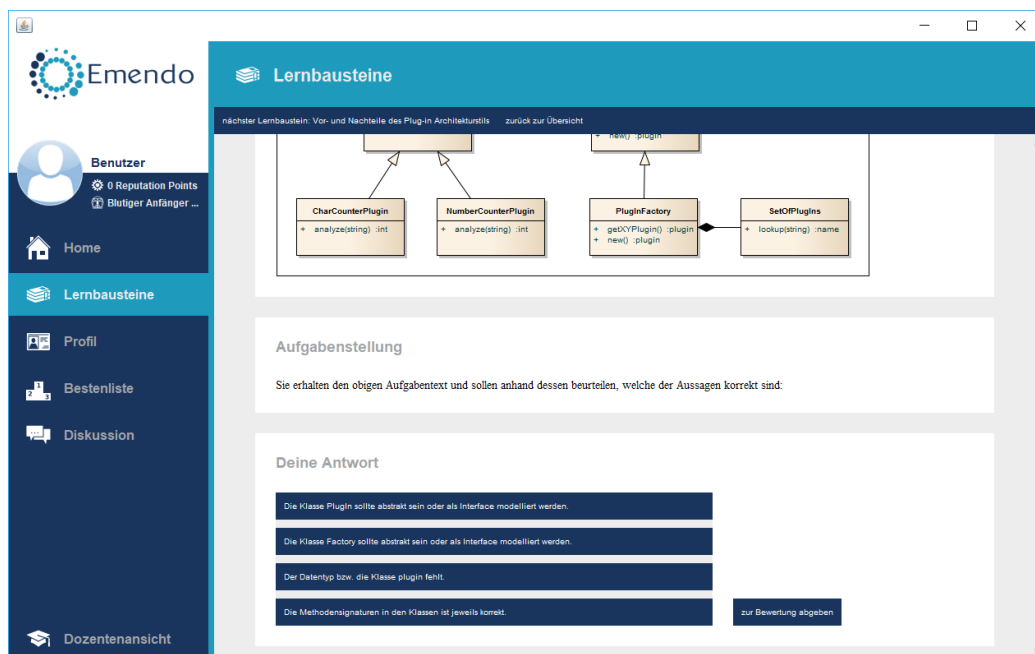


Abbildung 103: Screenshot Bearbeitung Multiple Choice-Aufgabe

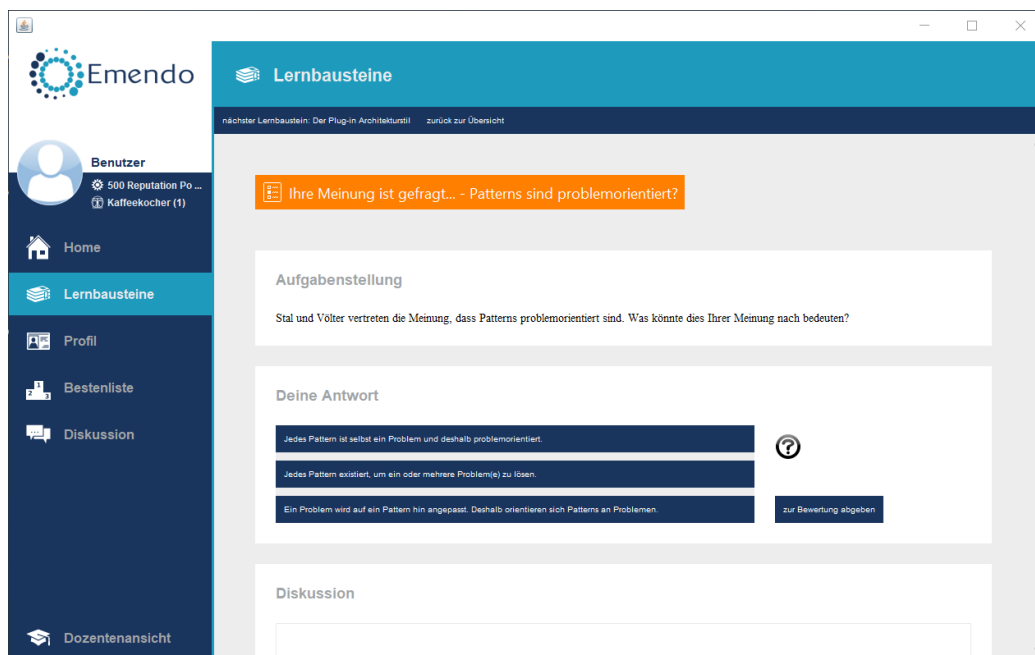


Abbildung 104: Screenshot Bearbeitung Single Choice-Aufgabe

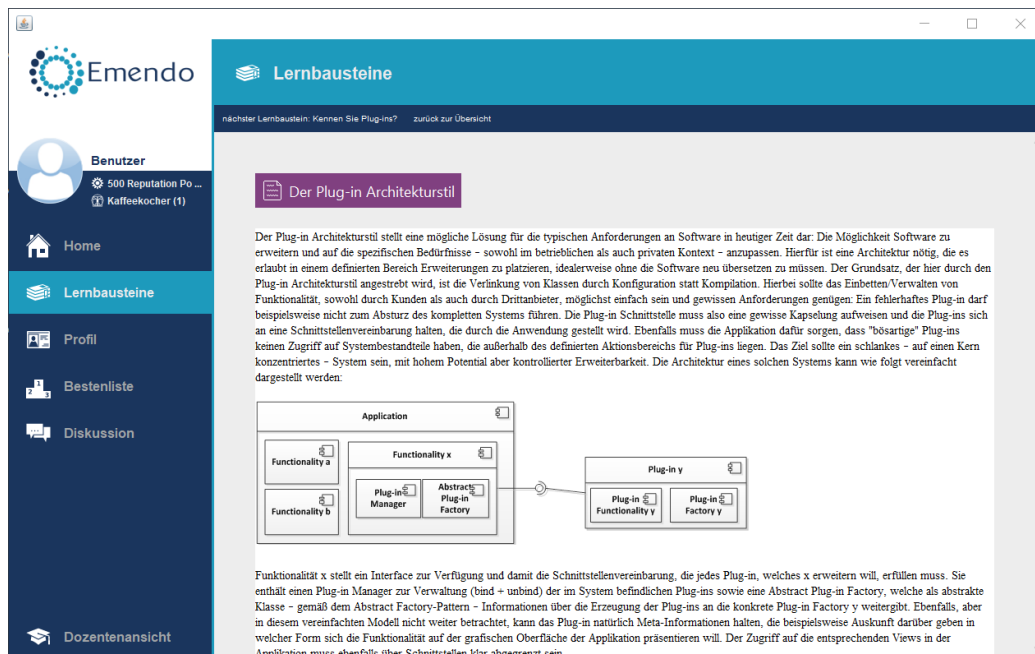


Abbildung 105: Screenshot Bearbeitung eines Textblocks

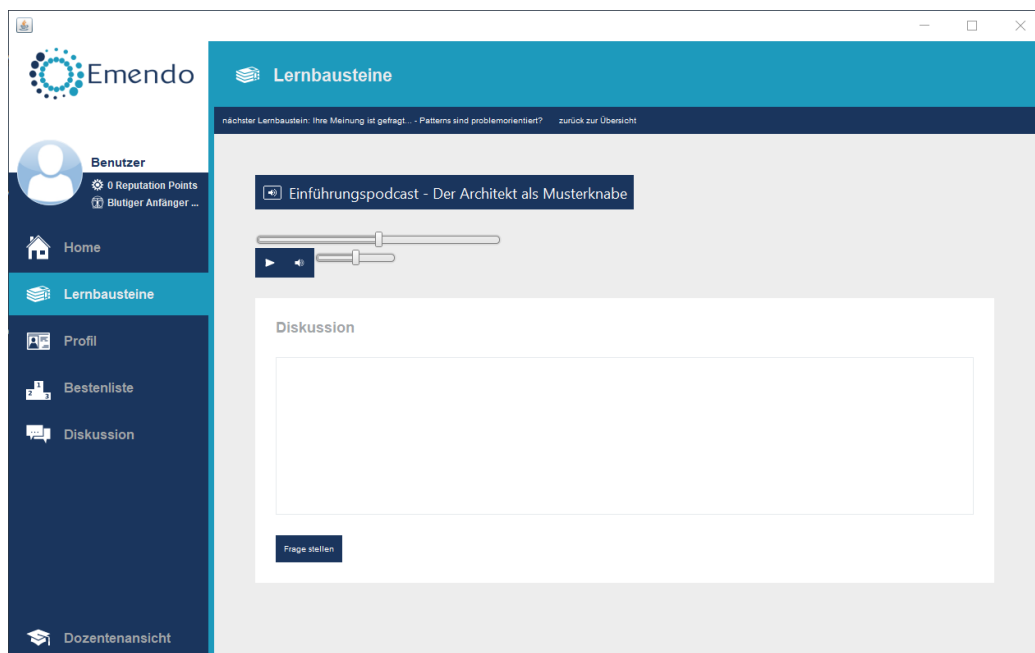


Abbildung 106: Screenshot Bearbeitung eines Podcasts

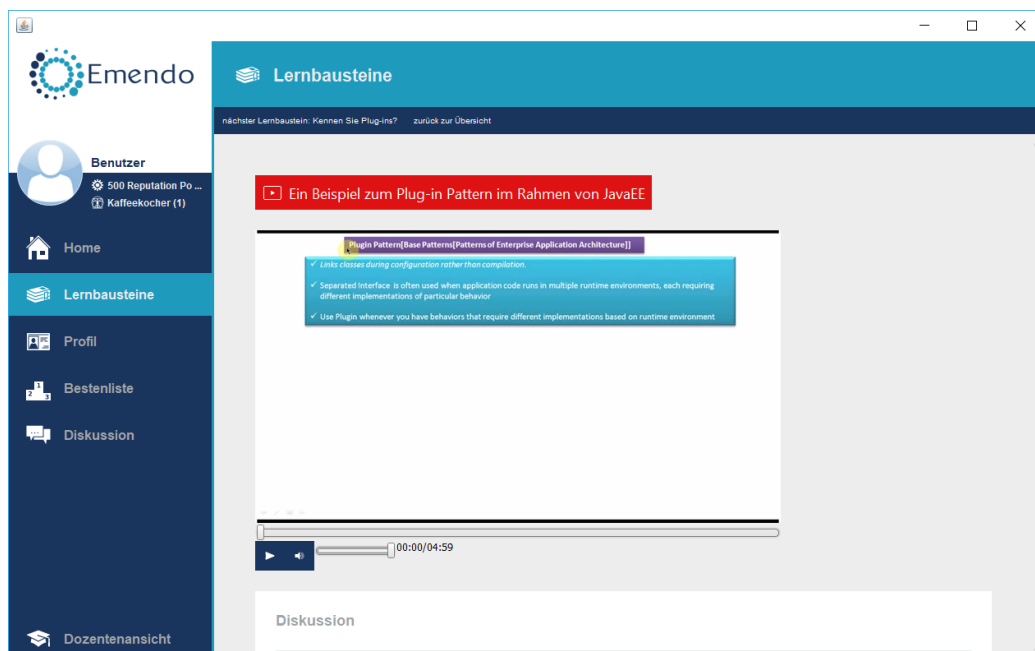


Abbildung 107: Screenshot Bearbeitung eines Videos



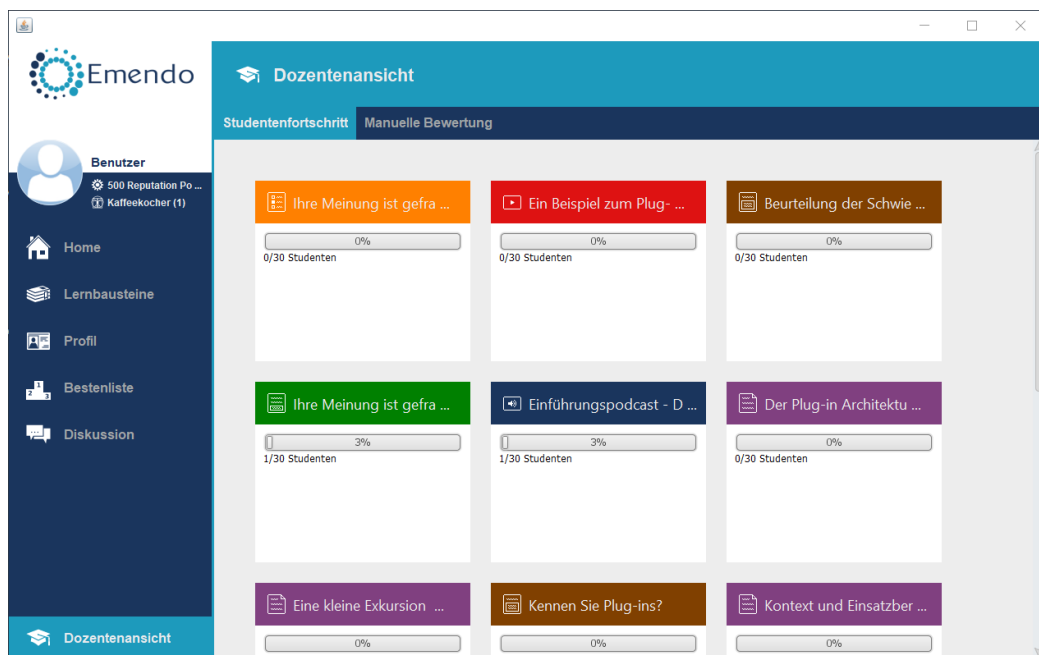


Abbildung 108: Screenshot Dozierendenansicht zum Lernfortschritt

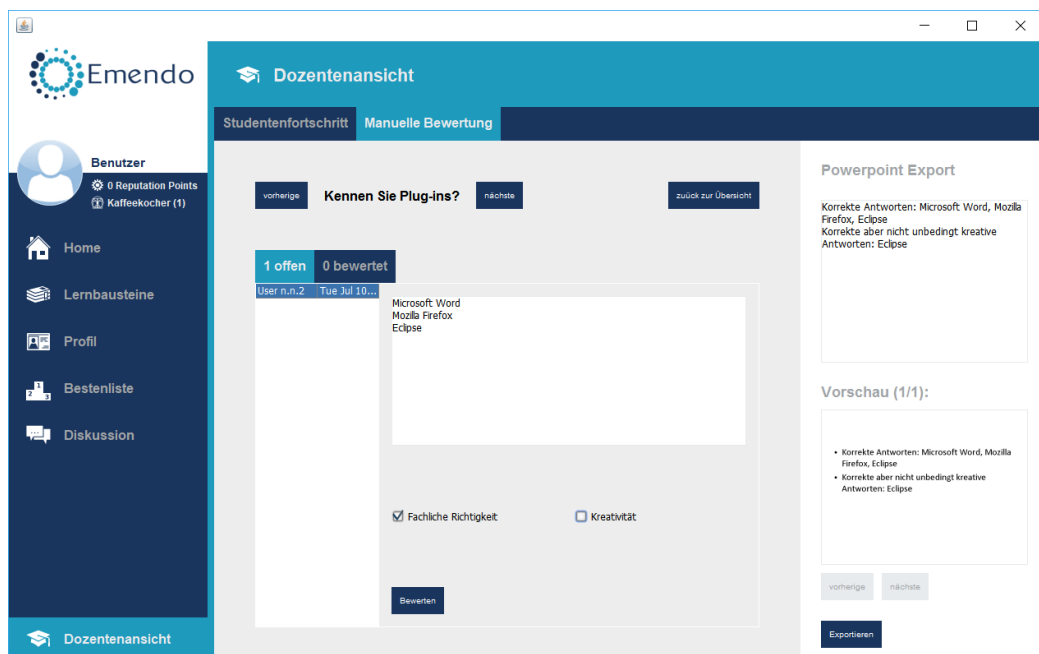
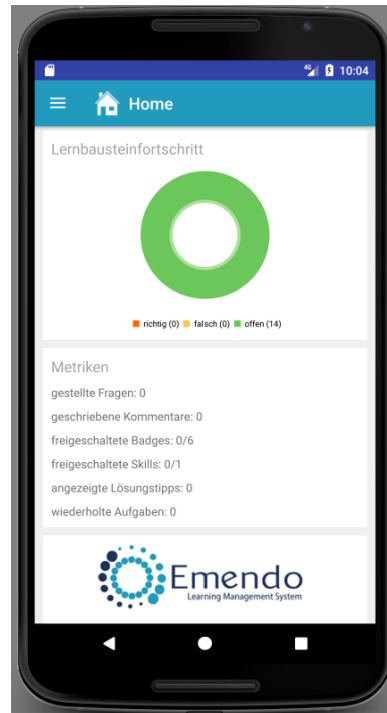


Abbildung 109: Screenshot Dozierendenansicht manuelle Bewertung

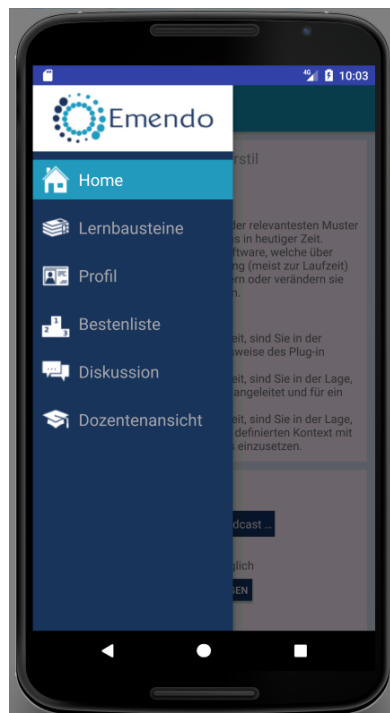
## A.7 SCREENSHOTS DES EMENDO LMS MOBILE



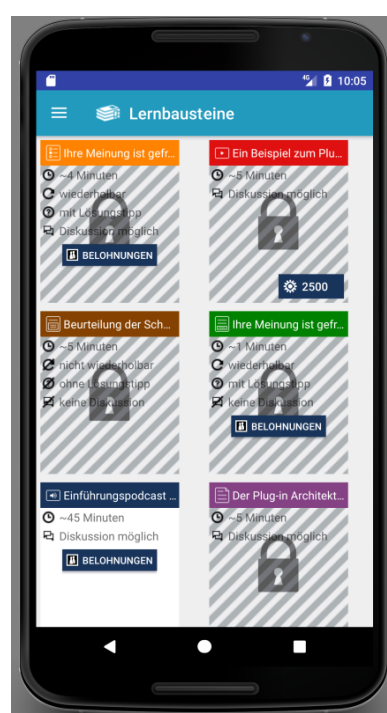
(a) Homescreeen 1



(b) Homescreeen 2

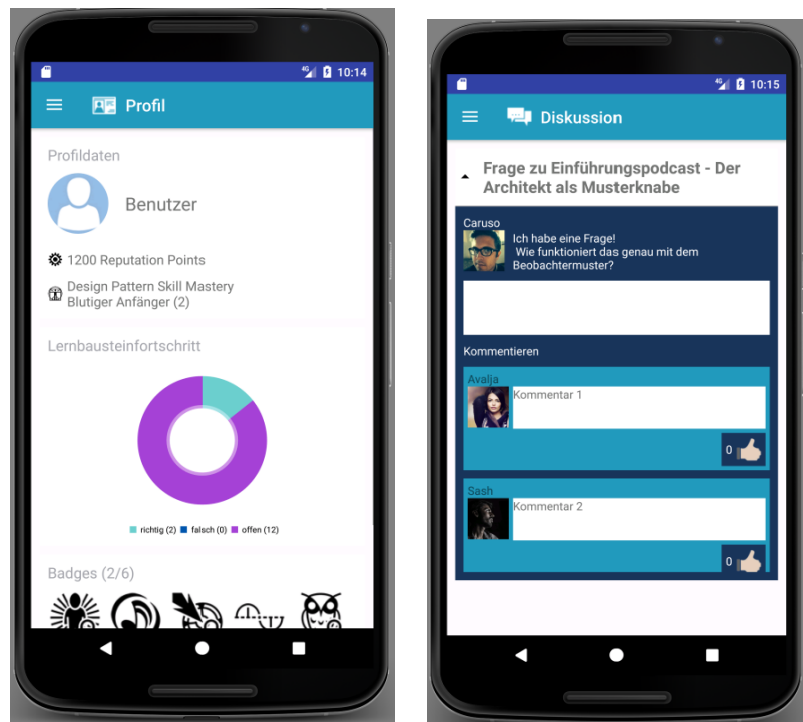


(c) Menu



(d) Lernbausteinübersicht

Abbildung 110: Screenshots des Emendo LMS Mobile – Hauptseiten



(a) Profilansicht

(b) Diskussionsbereich



(c) Bestenliste

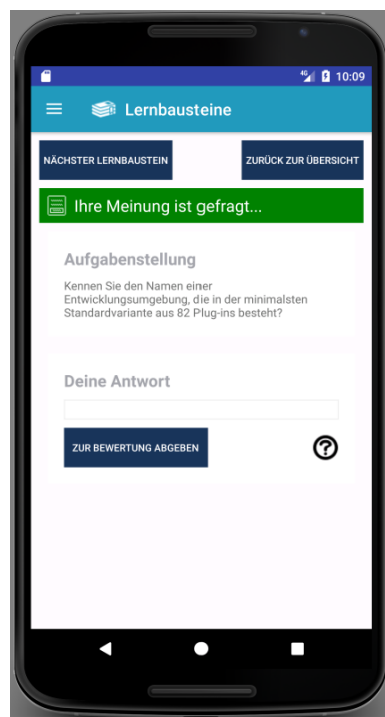
Abbildung 111: Screenshots des Emendo LMS Mobile – Hauptseiten Fortsetzung



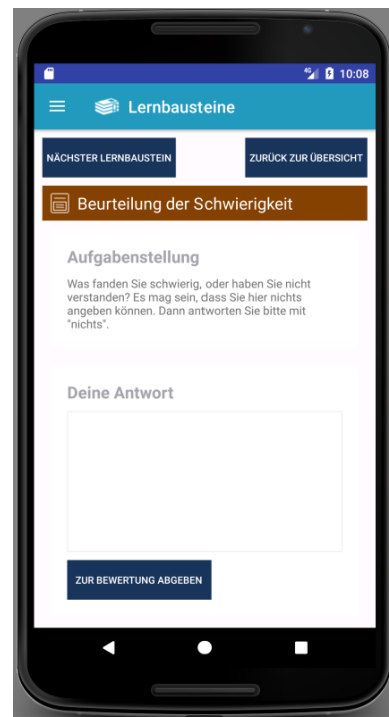
(a) Single Choice-Aufgabe



(b) Multiple Choice-Aufgabe

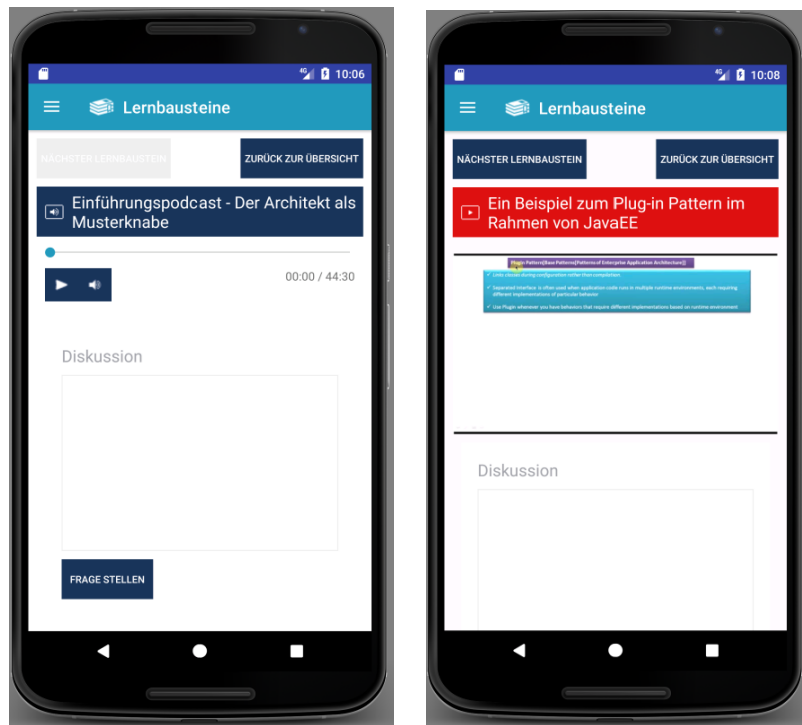


(c) Eingabe-Aufgabe



(d) Freitext-Aufgabe

Abbildung 112: Screenshots des Emendo LMS Mobile – Aufgabentypen



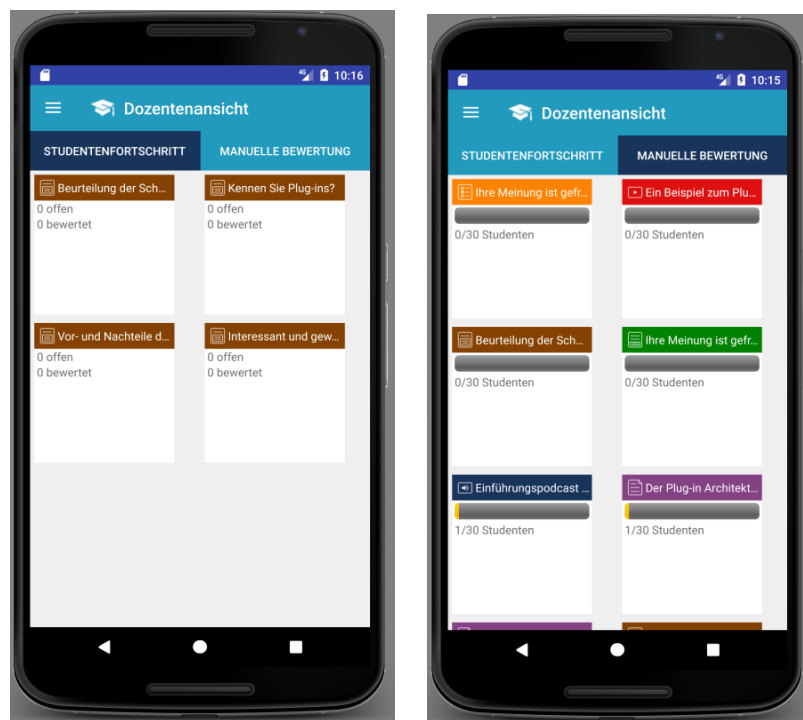
(a) Podcast

(b) Video



(c) Textblock

Abbildung 113: Screenshots des Emendo LMS Mobile – Lernmaterialien



(a) Dozentenansicht Manuelle  
Bewertungsübersicht

(b) Dozentenansicht Lernfort-  
schritt

Abbildung 114: Screenshots des Emendo LMS Mobile – Dozentenansicht

## A.8 GEHEFT FÜR DIE ZWEITE EVALUATIONSSTUFE

## Modellieren im Emendo Designer

Sie haben Emendo und seine Werkzeuge bzw. Bestandteile bereits kennengelernt und Ihre Meinung zu dem allgemeinen Konzept von Emendo geäußert. Nun soll es darum gehen, dass Sie als potentielle spätere Anwender erste Erfahrungen mit dem Emendo Designer sammeln, indem Sie die Emendo DSML benutzen, um eine spielifizierte Lerneinheit zu modellieren.

Im Folgenden finden Sie eine Beschreibung zur Vorbereitung auf die Modellierung sowie im Anschluss eine Szenariobeschreibung einer spielifizierten Lerneinheit, die Sie bitte im Emendo Designer als Modell abbilden. Sollten Sie während der Umsetzung Fragen oder Anmerkungen haben, äußern Sie diese bitte frei heraus.

### Schritte zur Vorbereitung:

1. Öffnen Sie das Projekt mit dem Namen *Evaluation* im Emendo Designer. Sie finden darin ein vorbereitetes Projekt mit mehreren Dateien, die technisch zur Modellierung notwendig sind.
2. Navigieren Sie in der Datei *representations.aird* zu dem Modell *EvaluationsLerneinheit* und öffnen Sie dieses, um mit der Modellierung zu beginnen. Jedes Modell enthält eine Gruppe (*Wurzelgruppe*), in die Lernelemente platziert werden können.

### Allgemeine Definition der Lerneinheit:

Für jede Lerneinheit müssen allgemeine Daten angegeben werden, die der Ersteller, Lernziele, etc. Hierzu klicken Sie im Emendo Designer auf das schwarze Zahnrad in der Menüleiste und geben die Daten in die entsprechenden Formularfelder ein.

### Hinweise:

Treffen Sie für die Inhalte der einzelnen Lernelemente selbst bitte Annahmen. Sollte Ihnen ein Inhalt fehlen, wie beispielsweise ein Bild oder ein Link zu einem Video, übernehmen sie dies bitte aus der im Projekt *Evaluation* befindlichen Textdatei *Mediensammlung.txt*. Für die Eingabe von Inhalten nutzen Sie bitte die Dialoge, die sich jeweilig bei einem Doppelklick auf ein Symbol öffnen.

Für die Modellierung stehen Ihnen in diesem Geheft diverse Hilfsmittel zur Verfügung. Sie finden neben einer Übersicht und Beschreibung der Notationselemente der Emendo DSML auch eine Matrix, die zeigt, welche Verbindungen zwischen Notationselementen zulässig sind.

Szenario:

Lernende sollen zu Beginn ein Video ansehen, das ein Thema einführt. Nach dem Video haben Lernende die Optionen, zunächst eine Single Choice Aufgabe zu beantworten oder einen Text lesen zu, der an die Inhalte des Videos anknüpft. Sollten sie die Single Choice Aufgabe richtig beantworten, erhalten Lernende dafür einen Badge als Belohnung. Wird sie falsch beantwortet, sollen Lernende ein Feedback erhalten, das bei der korrekten Lösung der Single Choice Aufgabe hilft, die wiederholt werden kann.

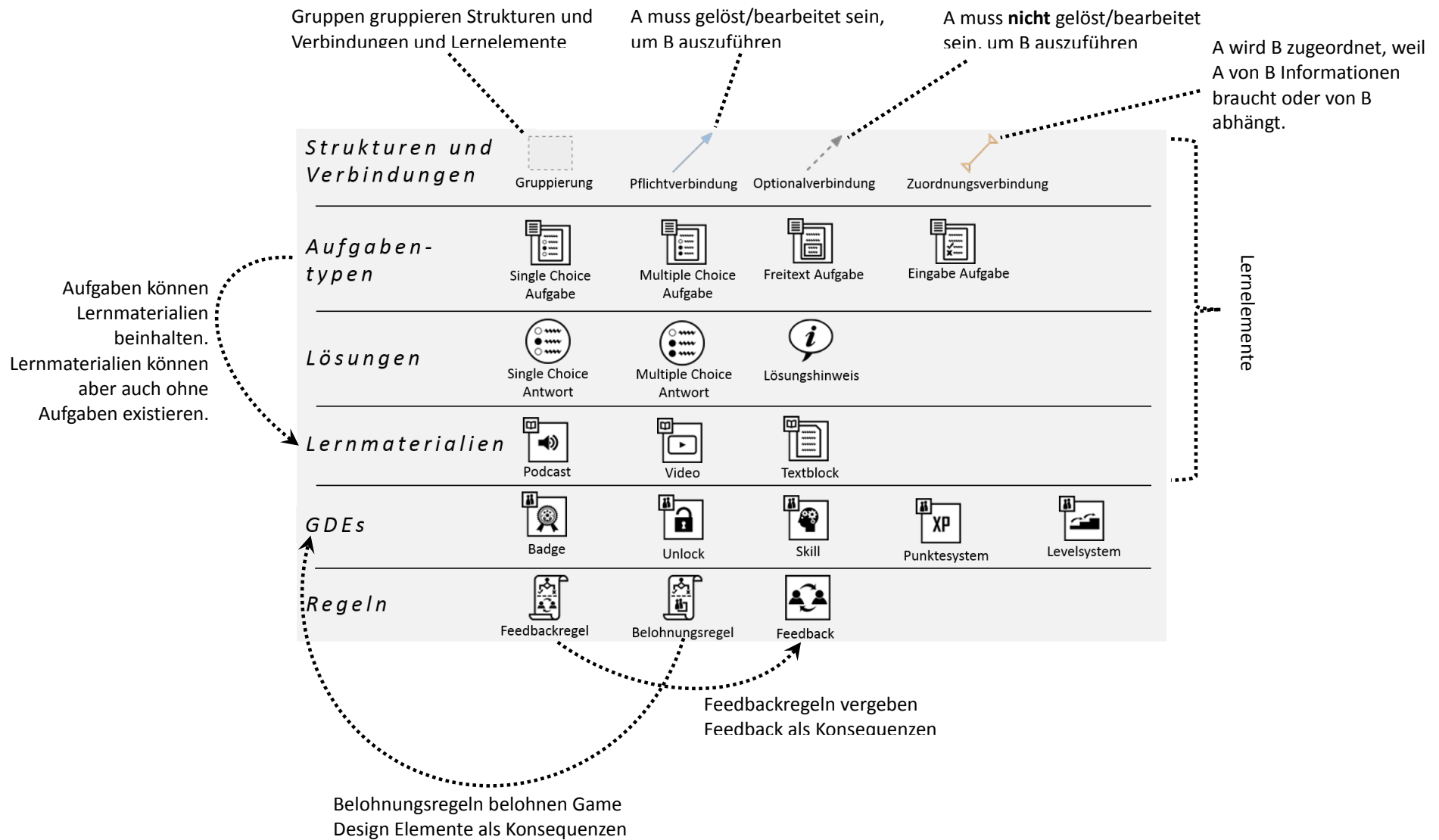
Im Anschluss an die Single Choice Aufgabe müssen Lernende einen Text lesen. Letzterer stellt auch den nächsten Pflichtteil des zuvor gelesenen Textes dar. Nach dem Lesen des Textes sollen Lernende eine Freitextaufgabe bearbeiten, wobei deren Antworten durch 3 Kriterien von Lehrenden bewertet werden soll. Der Freitextaufgabe ist ein Lösungshinweis zugeordnet, der bei der Lösung der Aufgabe hilft, daher soll sie nicht wiederholbar sein. Sind mehr als 60% der Kriterien erfüllt, gilt die Freitextaufgabe als korrekt beantwortet. Wird sie korrekt beantwortet, erhalten Lernende dafür als Belohnungen 5000 Punkte und einen Badge, den sie noch nicht erspielt haben.

Nach der Freitextaufgabe müssen Lernende eine Single Choice Aufgabe bearbeiten, die 3 Antwortmöglichkeiten hat, von denen jedoch nur eine richtig ist. Es dürfen keine Fragen zu der Aufgabe gestellt werden, jedoch darf sie wiederholt werden.

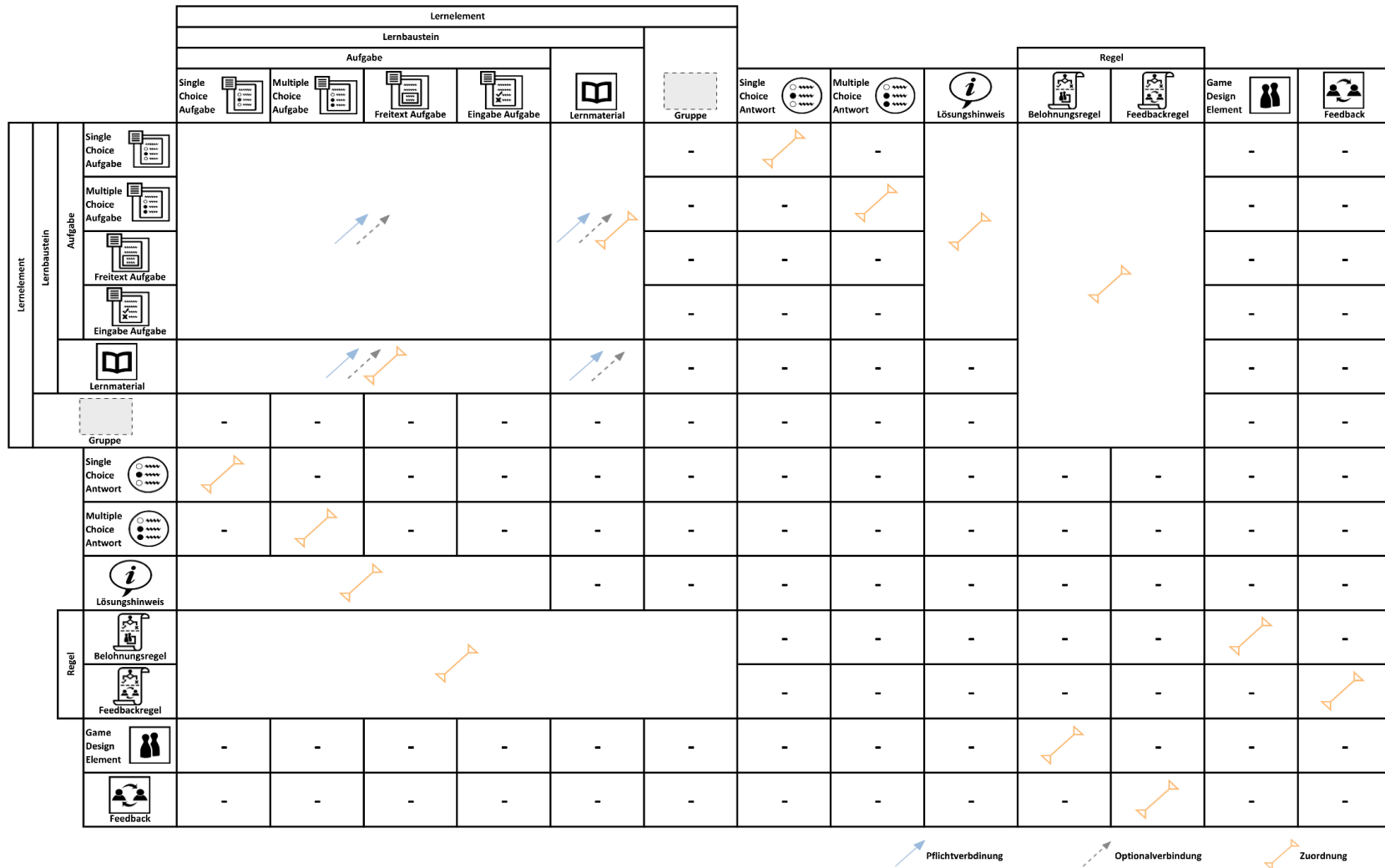
Werden alle Aufgaben innerhalb der Wurzelgruppe korrekt beantwortet, sollen Lernende mit einem Unlock belohnt werden, dass eine Eingabeaufgabe zur Bearbeitung für Lernende freischaltet.

Das Szenario ist hier beendet. Sie haben nach einer kurzen Befragung die Möglichkeit, dieses Szenario selbst weiterzuentwickeln.





## Verbindungsübersicht über die Elemente der Emendo DSML



## A.9 FRAGENBOGEN FÜR EVALUATIONSTEILNEHMENDE

## Fragebogen zur Evaluation von Emendo

Liebe Teilnehmende,

herzlichen Dank für Ihre Teilnahme an der Evaluation. Als Ergänzung zur mündlichen Befragung werden einige Angaben zu Ihrer Lehrtätigkeit und Ihrer Person in Form eines schriftlichen Fragebogens benötigt. Die Angaben werden, wie das Gespräch auch, anonymisiert ausgewertet. Zu keinem Zeitpunkt ist es möglich auf Ihre Person Rückschlüsse zu ziehen.

**Definition Gamification:** „Gamification ist ein Konzept, welches spielerische Elemente und Prozesse in Lernaktivitäten integriert, um die Lernmotivation zu erhöhen und das Verhalten der Lernenden dadurch ändert.“

**Wie häufig haben Sie Elemente von Gamification in Ihrem Berufsfeld eingesetzt?**

Sehr  
wenig

☐ ☐ ☐ ☐ ☐ ☐ ☐

Sehr häufig

**Wenn Sie bereits Gamification eingesetzt haben, geben Sie hier bitte ein Beispiel an:**

**Wie häufig haben Sie Learning Management Systeme in Ihrem Berufsfeld eingesetzt?**

Sehr  
wenig

☐ ☐ ☐ ☐ ☐ ☐ ☐

Sehr häufig

**Wenn Sie bereits Learning Management Systeme eingesetzt haben, geben Sie hier bitte ein Beispiel an:**

**Lehrerfahrung**

- |  |  |                                       |
|--|--|---------------------------------------|
| <input type="checkbox"/> bis 2 Jahre   | <input type="checkbox"/> 2 – 5 Jahre   | <input type="checkbox"/> 6 – 10 Jahre |
| <input type="checkbox"/> 11 – 15 Jahre | <input type="checkbox"/> 16 – 20 Jahre | <input type="checkbox"/> ab 21 Jahre  |

**Höchster akademischer Grad**

- |  |  |   |
|--|--|---|
| <input type="checkbox"/> keiner (z.B. Tutor) | <input type="checkbox"/> Bachelor      | <input type="checkbox"/> Master / Magister / Diplom |
| <input type="checkbox"/> Doktor(in)          | <input type="checkbox"/> Professor(in) |   |

**Fachbereich des höchsten akademischen Grades**

- |   |                                    |                                     |
|---|------------------------------------|-------------------------------------|
| <input type="checkbox"/> Informatik                 | <input type="checkbox"/> Pädagogik | <input type="checkbox"/> Mathematik |
| <input type="checkbox"/> Anderer Fachbereich: _____ |                                    |                                     |

**Dienstliche Stellung**

- |   |   |  |
|---|---|--|
| <input type="checkbox"/> Tutor(in)          | <input type="checkbox"/> wiss. Mitarbeiter(in)            | <input type="checkbox"/> Akad. Rat/Rätin |
| <input type="checkbox"/> Lehrbeauftragte(r) | <input type="checkbox"/> Lehrkraft für besondere Aufgaben |  |
| <input type="checkbox"/> Professor(in)      | <input type="checkbox"/> Andere Position: _____           |  |

**Alter**

- |  |  |  |
|--|--|--|
| <input type="checkbox"/> bis 25 Jahre  | <input type="checkbox"/> 26 – 35 Jahre | <input type="checkbox"/> 36 – 45 Jahre |
| <input type="checkbox"/> 46 – 55 Jahre | <input type="checkbox"/> 56 – 65 Jahre | <input type="checkbox"/> ab 66 Jahre   |

Herzlichen Dank für Ihre Teilnahme an der Studie!

## ANMERKUNGEN ZUM ABSCHLUSS

### *Designempfehlungen in classicthesis*

Die vorliegende Arbeit wurde auf Basis der L<sup>A</sup>T<sub>E</sub>X-Vorlage `classicthesis` von Miede (2012) angepasst und angefertigt. Somit kommen in diesem Werk typographische Designempfehlungen von Bringhurst (2008) zum Einsatz.